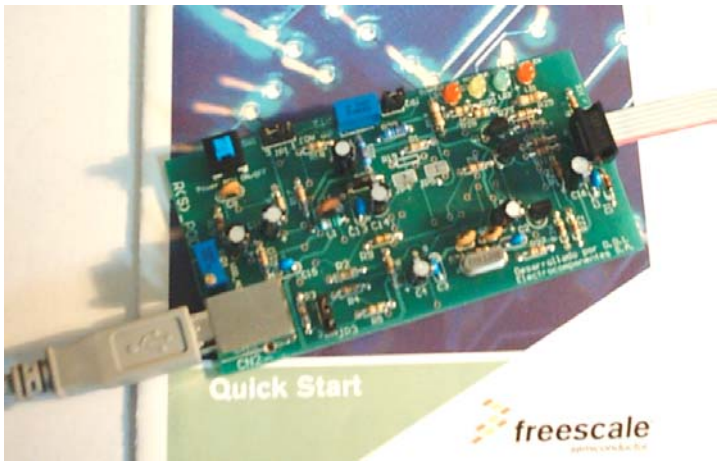
"R(S)_POD"



Manual del Usuario

Versión 1.1

Sistema de Emulación en Tiempo Real BDM / ICE para TODAS las familias de Microcontroladores Flash HC9S08, RS08, Serie Flexis 8 / 32 Bits de Freescale Semiconductor.

WWW.EDUDEVICES.COM.AR

freescale™ semiconductor

Control de versiones:

1.0 .- Edición ORIGINAL del 2007.

1.1 .- Editada en Septiembre 2008.

En esta **versión 1.1** se hace referencia a la **nueva versión** de **"Firmware" (versión R7)** disponible para actualizar la versión actualmente instalada en los sistemas ***R(S)_POD*** que están siendo utilizados en el mercado **(versión de Firmware R5).** Esta nueva versión permite trabajar con la familia de la **Serie Flexis ColdFire V1 (32 bits) presente y futura,** de la misma forma que se lo hacía con las familia HC9S08 / RS08 / Serie Flexis HC9S08.

Los sistemas fabricados a partir de la presente edición, contendrán el **Firmware Versión R7** como firmware por defecto de fábrica, por lo que **el presente manual es válido** para la **versión R7** de firmware del sistema ***R(S)_POD.***
Para usuarios de la **versión R5**, referirse al **Manual de Usuario Versión 1.0 (RS_POD_UM1V0.pdf)**

- Contenido.

1.0 – Introducción.
2.0 – Características Generales.
3.0 – Contenido del Kit.
4.0 – Instalación del Software.
5.0 – Instalación del Hardware del Sistema.
6.0 – Configuración de Jumpers / Switchs & LEDs.
7.0 – Conector BDM para la conexión con la tarjeta de usuario.
8.0 – Poniendo en Marcha el sistema.
9.0 – Instalación Drivers USB en Windows 98SE.
10.0 – Preguntas más Frecuentes.
11.0 – Actualización de Firmware.
Addendum1 : Jumpers & Switches Layouts
Addendum2 : Como utilizar los valores de fábrica de NVFTRIM / NVICSTRIM

Diseñado y Desarrollado por:

Ing. Daniel Di Lella

Dedicated Field Application Engineer (D.F.A.E)
(Ingeniero de Aplicaciones)

EduDevices.com.ar

E-Mail Privado: dilella@arnet.com.ar

Comentarios & Sugerencias:

EduDevices.com.ar
República Argentina
Tel: (54 11) 4584 – 3142
edudevices@gmail.com
www.edudevices.com.ar

1.0 - Introducción:

El propósito de este manual es describir el funcionamiento del sistema ***R(S)_POD*** y dar instrucciones paso a paso que permitan el uso de esta potente herramienta de desarrollo para los nuevos microcontroladores de la familia **HC9S08 / RS08 / Serie Flexis de 8 / 32 Bits** de Freescale Semiconductor.

El sistema ***R(S)_POD*** es una herramienta de desarrollo para diseñadores que desarrollan y depuran tarjetas que incorporan alguno de los microcontroladores (MCU) **HC9S08xx / RS08xx / Serie Flexis HC9S08 / ColdFire V1** y el mismo está basado en el sistema de emulación abierto **"OSBDM"** desarrollado por la comunidad Freescale como herramienta de bajo costo.
El KIT contiene todo el hardware y software necesario para desarrollar una aplicación completa, incluyendo, **Emulación En Circuito en verdadero Tiempo Real**, **Simulación Pura sin circuito, Edición y Compilación de Código y Programación de los microcontroladores de la familia HC9S08 / RS08 / Serie Flexis HC9S08 / ColdFire V1.**

El sistema ***R(S)_POD*** ha sido concebido como una herramienta de muy bajo costo con características especiales para el mercado local, bajo ningún punto de vista pretende reemplazar la funcionalidad y flexibilidad universal del sistema **USBMULTILINKBDME** diseñado y fabricado por P & E Microcomputer Systems para Freescale Semiconductor. Solo el sistema universal USBMULTILINKDBME asegura una total compatibilidad con todos los dispositivos presentes y futuros HC9S08 / RS08 / Serie Flexis HC9S08 / ColdFire V1.

A título ilustrativo, se detalla una **lista parcial** de los MCUs soportados por el sistema ***R(S)_POD:***

- Familia MC9S08GB
- Familia MC9S08GT
- Familia MC9S08AW / AC
- Familia MC9S08QG / QA / QD
- Familia MC9S08RC
- Familia MC9S08SH
- Familia MC9S08DZ
- Familia MC9RS08KA1 / KA2 / KA8
- Serie Flexis HC9S08 (QE, AC, JM, etc.)
- Serie Flexis ColdFire V1 (QE, AC, JM, etc.)

2.0 Características Generales:

Antes de definir las características que posee el sistema ***R(S)_POD*** deberemos entender las particularidades y modo de funcionamiento de los Modos BDM y On – Chip ICE disponibles en la familia de microcontroladores **HC9S08 / RS08 / Serie Flexis HC9S08 / ColdFire V1.**

El módulo ***BDM (Background Debug Mode)*** provee una interface a un solo hilo que permite leer y escribir la memoria del MCU, examinar y cambiar los registros del CPU, establecer los breakpoints (puntos de parada) y depuración paso a paso por medio de un programa específico.
Este tipo de interface es común en la mayoría de los microcontroladores más modernos existentes hoy en día.

El segundo sistema, es el ***On – Chip In Circuit Emulator (On – Chip ICE)***, es un Emulador en Tiempo Real en Circuito integrado al chip equivalente a los Emuladores Externos tradicionales con breakpoints por hardware, disparos lógicos (logic triggers), y un buffer de captura de Bus que puede capturar direcciones o datos relativos a los eventos de disparo. El ***On – Chip ICE*** captura información del Bus sin interferir con el programa de aplicación de la misma forma como lo haría un analizador lógico, con la excepción de no necesitar un circuito externo de alto costo.

El sistema On – Chip ICE, no es un módulo muy común en la mayoría de los microcontroladores de otras marcas. Ello constituye un avance significativo que permite reemplazar completamente el uso de sistemas de emulación externos. Este módulo viene incorporado en TODOS los microcontroladores de las familias HC9S08 / RS08 / Serie Flexis HC9S08 / ColdFire V1 de Freescale Semiconductor.

El módulo de BDM de las familias HC9S08 / RS08 / Serie Flexis 8 / 32 Bits, es del mismo tipo que el utilizado en primer lugar por la familia HC9S12. Como veremos más adelante, desde el punto de vista del hardware, el sistema ***R(S)_POD*** es totalmente compatible con la familia HC9S12, pero además se necesita un programa de interface de software (del tipo General Device Interface (GDI)) para vincular el sistema (POD) con el entorno de desarrollo.

Este BDM es un avance sobre otras interfaces seriales de depuración, porque el BDM utiliza solo un pin de microcontrolador bajo desarrollo (BKGD). El protocolo implementado, le permite al host (PC) detectar y adaptar la velocidad de comunicación a la velocidad del sistema bajo desarrollo. El mismo permite un error de sincronización de un +/- 10% a +/- 20% entre la velocidad de la PC y de la placa con el MCU, de esta forma, las comunicaciones son muy robustas.

Características del R(S)_POD:

Características propias de los módulos BDC y ICE DBG:

- Utiliza un solo pin (hilo) para selección de Modo y la comunicación en "Background" con el MCU.
- Los registros del módulo BDC no están alojados en el mapa de memoria.
- Comando "SYNC" para determinar la velocidad de comunicación con la placa bajo desarrollo.
- Comandos "No – Intrusivos" para acceder a los espacios de memoria.
- Comandos "No – Intrusivos" significa que el MCU ejecuta el programa del usuario sin demoras de tiempo mientras en forma paralela se ejecutan dichos comandos.
- Comandos para "Modo Background Activo" que permiten modificar los registros del CPU y tener el control total del MCU.
- Comandos "GO" y "TRACE1".
- Comando "BACKGROUND" puede despertar al CPU desde los modos de STOP y WAIT.
- 1 Hardware Breakpoint (punto de parada) por dirección de PC integrado al módulo BDC.
- El oscilador "corre" en el modo STOP cuando el módulo BDC está habilitado.
- COP "Watchdog" deshabilitado mientras el MCU está en modo "Background. Activo".
- 2 comparadores de disparo (Trigger Comparators):
  - 2 direcciones + Lectura / Escritura (R/W) o
    1 dirección completa + dato + R/W.
- Buffer flexible de captura de información de 8 palabras de 16 bits c/u del tipo FIFO con modalidad de captura "cambio de flujo en el direccionamiento" o "Dato de evento solo".

- 2 tipos de Breakpoints:
  - "Tag Breakpoints" para códigos de instrucciones.
  - "Force Breakpoints" para cualquier dirección de acceso.
- 9 modos de disparo.

Características propias al sistema R(S)_POD:

- Basado en el ***Open Source BDM (OSBDM)*** desarrollado por la "comunidad Freescale" como herramienta de bajo costo.
- Permite Borrar / Grabar la memoria FLASH de los MCUs de la familia HC9S08 y RS08.
- **Verdadera Emulación de Tiempo Real del tipo BDM / ICE en circuito**.
- Maneja Frecuencias de Bus del MCU bajo desarrollo desde **1 MHz a 25 MHz.**
- Detección automática de la frecuencia del Bus y velocidad de comunicación.
- Emulación a un solo Hilo (BKGD).
- Permite manejar el pin de RESET (RST) del MCU bajo desarrollo, para aquellos dispositivos que así lo dispongan (Control por Hardware y por Software).
- Pin **"RST/Vpp"** que permite el borrado / grabación de la memoria FLASH de los dispositivos de la familia **RS08** al proveer la tensión de Vpp (+12V) necesaria para ello (manejo automático del mismo).
- Soporta tensiones de VDD del MCU bajo desarrollo desde **+1,8V a +5Vdc.**
- Dos modos de Alimentación de la interface lógica:
  - Toma alimentación desde la placa bajo desarrollo (VDD del MCU).
  - Provee alimentación a la placa bajo desarrollo (VDD ajustable desde +1,8V a +5V).
- Sistema alimentado por puerto **USB 2.0** estándar de PC o Notebook.
- Soporta 300 mA de Corriente máxima hacia la placa de usuario bajo desarrollo.

- Protección activa por sobre corriente con indicación lumínica (Overcurrent) que evita daños al puerto USB de la PC o Notebook y al resto del circuito del usuario.

- Indicaciones lumínica de:

  - Actividad del puerto USB ---- **"USB_EN".**
  - Tensión presente en VDD del MCU ---- **"Power_On".**
  - Tensión +Vpp activa (pin RST/Vpp) --- **"Vpp_EN".**
  - Sobre Corriente en VDD ---- **"OverCurrent".**

- **Preparado para actualizaciones futuras de firmware del sistema, por parte del usuario, por medio del puerto USB.**

- Soporta entornos de trabajo integrados (IDE) como el ***CodeWarrior 5.0*** / ***5.1*** y el ***CodeWarrior 6.x*** para **HC9S08 / RS08 / Serie Flexis HC9S08 / ColdFire V1**.

- Sistema Operativo ***Windows XP SP2 / SP3*** (con algunas precauciones puede llegar a correr sobre plataformas ***Windows 98SE***).

3.0 – Contenido del KIT.

El kit del sistema ***R(S)_POD*** viene provisto de los siguientes elementos:

- **1 Placa *"R(S)_POD"* (placa principal del sistema).**
- **1 Placa *"Demo_QG8"* (placa demostrativa con MC9S08QG8 para realizar las primeras experiencias).**
- **1 Cable Serial USB tipo "A – B".**
- **1 CD ROM de instalación y versión Demo CodeWarrior 5.0**
- **1 CD ROM de versión Demo CodeWarrior 6.2**
- **Hojas de Instrucciones de instalación Rápida.**

4.0 - Instalación del Software.

En la actualidad, existen varias versiones del sistema ***CodeWarrior*** de Freescale Semiconductor disponibles para utilizar con las familias de microcontroladores Flash de 8 Bits como **HC908, HC9S08, RS08, Serie Flexis HC9S08 / ColdFire V1 de 32 Bits.**

El Kit del sistema ***R(S)_POD*** se provee con las versiones "Demo" de los sistemas ***CodeWarrior 5.0*** y ***CodeWarrior 6.2*** en forma gratuita (CD ROM´s de cortesía) con las siguientes limitaciones:

- **CodeWarrior 5.0:** apto para MCUs de la familia HC908 y HC9S08, puede instalarse en PCs o Notebooks con sistemas operativos Windows 98SE / Me / XP y no requiere de grandes exigencias del hardware de la PC, con procesadores Pentium III o similar y 128 Mbytes de RAM es suficiente. La versión gratuita permite compilar sin límites código assembler y hasta 16 Kbytes en código "C". **Esta versión solo es recomendada para PC´s antiguas, con muy pocos recursos, ya que no soporta muy bien herramientas como el *R(S)_POD* o similares.**

- **CodeWarrior 5.1:** apto para MCUs de la familia HC908, HC9S08 y RS08, puede instalarse en PCs o Notebooks con sistemas operativos Windows XP o superior y requiere de procesadores Pentium IV o equivalentes y más de 256 Mbytes de RAM. Posee un muy buen nivel de actualizaciones y soporta muchas herramientas distintas. La versión **gratuita** permite compilar **sin límites código assembler** y hasta **32Kbytes en código "C".**

- **CodeWarrior 6.0/6.1/6.2:** Son las últimas versiones disponibles, aptas para MCUs de la familia HC908, HC9S08, RS08 y la nueva "Serie Flexis" de 8 / 32 Bits (S08 / V1 ColdFire). Puede instalarse en PCs o Notebooks con sistemas operativos Windows XP o superior y requiere de procesadores Pentium IV o superiores y más de 512 Mbytes de RAM.
  Posee un muy buen nivel de actualizaciones y soporta muchas herramientas distintas. La versión **gratuita** permite compilar **sin límites código assembler** y hasta **32Kbytes en código "C" para la familia HC9S08 y 64Kbytes para la familia Flexis HC9S08 / V1 ColdFire (versiones recomendadas).**

La ampliación de las prestaciones básicas de ambos sistemas es muy sencilla y pueden ser efectuadas por Internet. Toda ampliación fuera de las prestaciones básicas es con cargo, para mayor información consultar en:

Web Site Freescale: www.freescale.com/codewarrior

El sistema ***CodeWarrior*** es un Entorno Integrado de Desarrollo **(IDE)** muy poderoso, que reúne un compilador, un linkeador y un debugger de código assembler fuente, y que puede ser ampliada según las necesidades del usuario.
Las versiones **"5.0"** y **"6.x"** ofrecen ensamblado de código fuente en forma ilimitada (assembler) y proveen capacidades de Debugging muy interesantes aún para programadores adelantados.

Esta herramienta poderosa, combina un Ambiente de Desarrollo Integrado de Alta perfomance (I.D.E) con:

- Simulación Completa del Chip y programación de la memoria FLASH y Verdadera Emulación en Tiempo Real desde el sistema ***R(S)_POD***.

- Un Compilador ANSI C (16K Bytes de código) altamente optimizado y un Debugger en nivel fuente C, Generación automática de código C con "Processor Expert", que permite una muy fácil inicialización de los distintos periféricos del MCU elegido.

Requerimientos Mínimos del Sistema (Versión CW 5.0):

Hardware:

- Procesador → Pentium II 400 Mhz o AMD – K6.
- Memoria RAM → 128 MB
- Puerto USB 2.0 (@ 500 mA.)
- Lectora de CD ROM.

Nota: El sistema ***CodeWarrior 6.x*** requiere prestaciones algo **superiores** tanto en el Hardware como en el sistema operativo para funcionar correctamente (ver párrafos anteriores).

Sistema Operativo:

- **Microsoft Windows 98 SE / 2000 / XP SP2 o SP3 (recomendados).**

Espacio en Disco:

- Instalación Compacta → 500 MB.
- Instalación Full → 635 MB.
- Actualizaciones posteriores → 1GB.

Como en todas las versiones de ***CodeWarrior*** los pasos de instalación son muy similares, procederemos a instalar la **Versión 5.0** para nuestro ejemplo.....

Se sugiere, de ser posible, **instalar la versión CW 6.2**, ya que esta dispone de mejor actualización vía Internet y además incluye el **"service pack"** para soporte integral de la herramienta **OSBDM** (Open Source BDM) en la cual está basado el sistema ***R(S)_POD.***

Pasos a seguir:

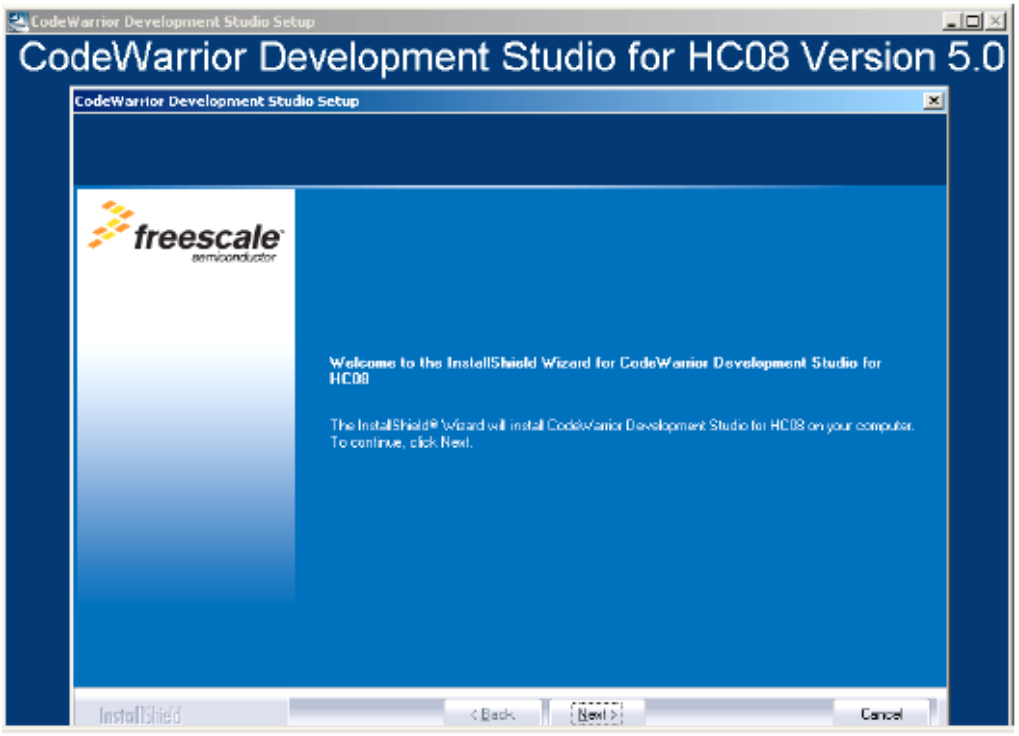
1) Introduzca el CD ROM con la etiqueta electrónica ***"R(S)_POD"*** en el lector de la PC o Notebook a utilizar y seleccione la siguiente ruta:

   D:\ R(S)_POD\CodeWarrior 5.0

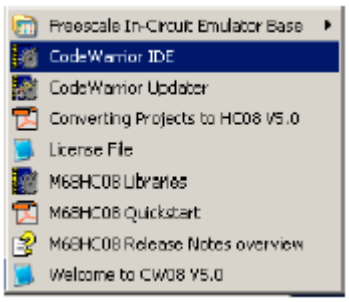
2) Ejecutar el archivo "Launch" de dicho directorio y nos aparecerá la siguiente pantalla .....

3) Elegir la opción "Launch the Installer" para comenzar con la instalación automática.

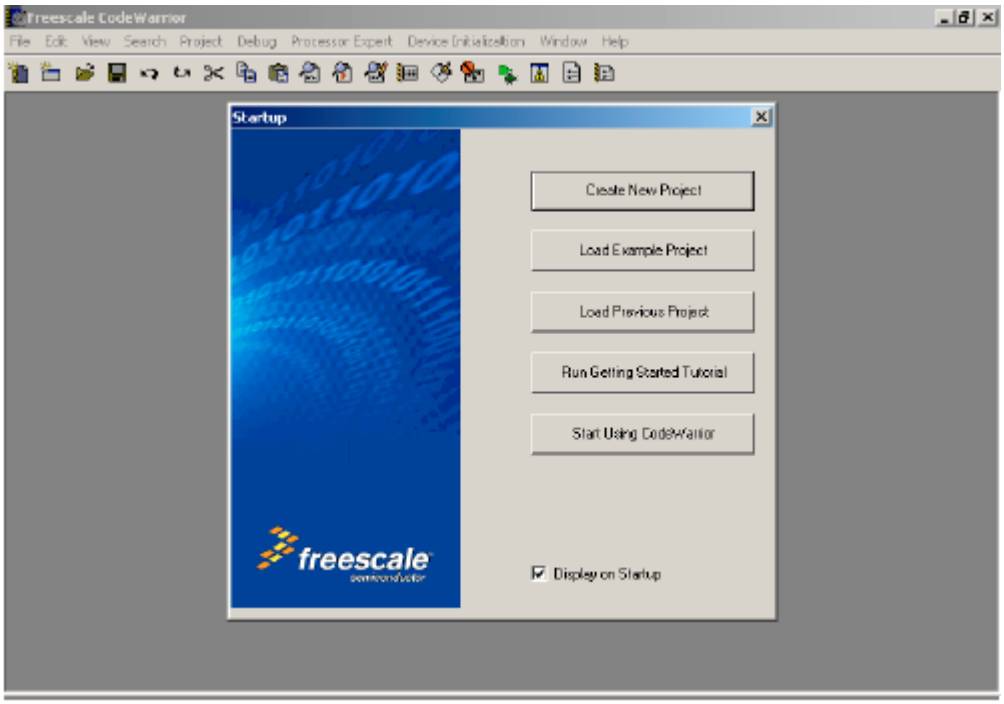
4) Aparecerá una pantalla de Bienvenida y se deberá seguir atentamente las indicaciones de las sucesivas pantallas a lo largo de la instalación que nos irán guiando paso a paso en la misma.


CodeWarrior Development Studio Setup
CodeWarrior Development Studio for HC08 Version 5.0
CodeWarrior Development Studio Setup
freescale
semiconductor
Welcome to the InstallShield Wizard for CodeWarrior Development Studio for HC08
The InstallShield® Wizard will install CodeWarrior Development Studio for HC08 on your computer. To continue, click Next.
InstallShield
< Back
Next >
Cancel

5) Una vez concluida la instalación, procederemos a lanzar el CodeWarrior 5.0 IDE Seleccionando Inicio → Programas → Freescale CodeWarrior → HC08 V5.x → CodeWarrior IDE (como se ve en la figura, o bien instalando un icono de acceso directo del CodeWarrior IDE).

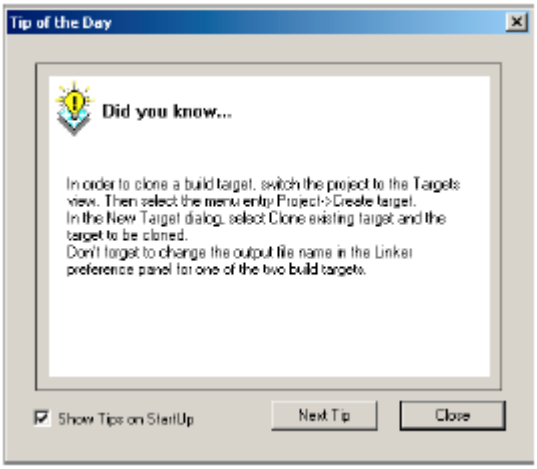


6) Haciendo "click" en el icono de acceso directo del escritorio o bien en la opción **"Codewarrior IDE"** de la ventana de programas mostrada anteriormente, se nos abrirá la siguiente pantalla de presentación.....



Siempre que se tenga "tildada" la opción **"Display on Startup"** se abrirá la ventana con opciones de comienzo de trabajo con el sistema que facilitarán nuestro aprendizaje en el mismo.

Al cerrar esta pantalla, se nos abrirá otra conteniendo **"tips"** que nos irán mostrando las distintas posibilidades que nos brinda este poderoso entorno.


Tip of the Day
Did you know...
In order to clone a build target, switch the project to the Targets view. Then select the menu entry Project->Create target. In the New Target dialog, select Clone existing target and the target to be cloned.
Don't forget to change the output file name in the Linker preference panel for one of the two build targets.
Show Tips on StartUp
Next Tip
Close

4.1 - Actualización del sistema CodeWarrior.

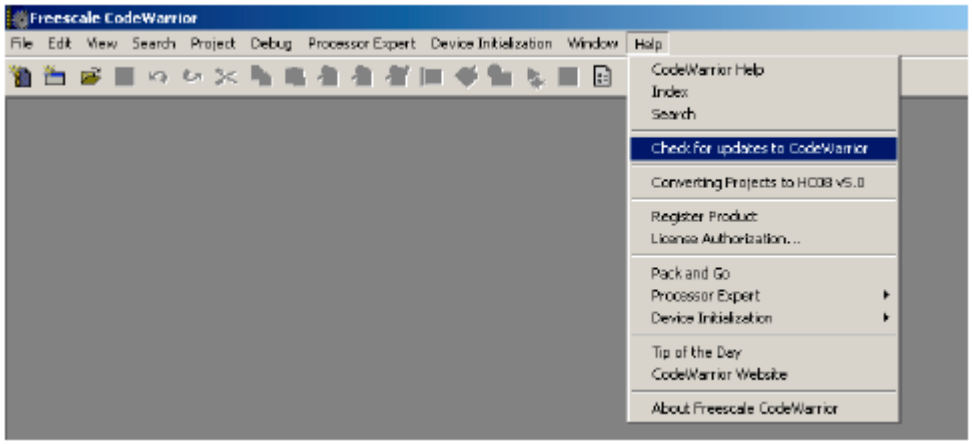
Los sistemas ***CodeWarrior*** disponen de la facilidad de actualización vía Internet lo que posibilita mantener al día nuestro sistema de desarrollo ante la aparición de nuevos dispositivos, correcciones o mejoras, ejemplos de aplicación, etc.

Es indispensable que en la PC o Notebook elegida para la instalación del sistema ***CodeWarrior*** se disponga de una conexión a Internet, ya sea en forma directa (modem) o bien por medio de la red en donde la misma se encuentre conectada.

Las versiones **demo** del ***CodeWarrior 5.0*** y ***6.x*** contenidas en los CD ROMs de cortesía, requieren de actualizaciones para funcionar correctamente con el sistema ***R(S)_POD***.

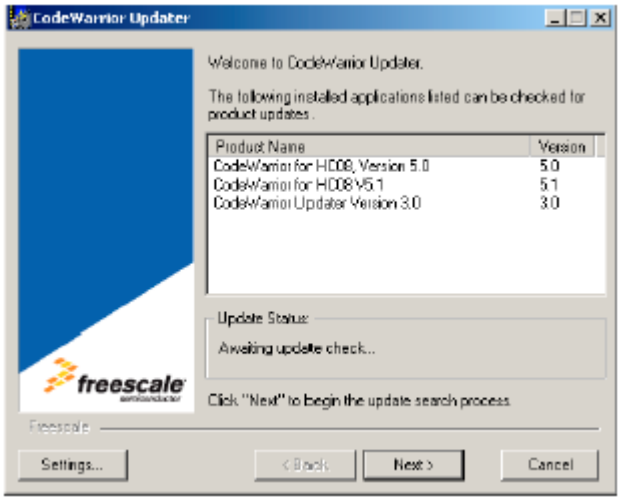
Para ello se procederá de la siguiente forma:

1) Ejecutar la aplicación **"CodeWarrior 5.0 / 6.x IDE".**
2) Cerrar la pantalla de presentación **"Startup"** y la pantalla **"Tip"** subyacente.
3) Hacer "click" en la opción **"Help"** de la **"Barra de Herramientas"** como se puede observar en la figura.

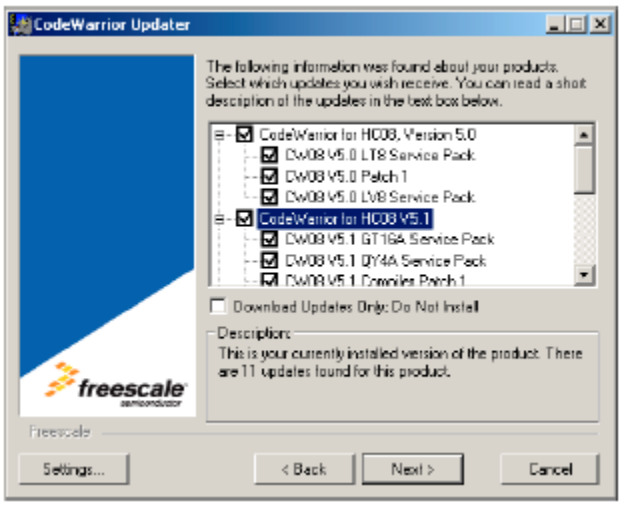


4) Hacer "Click" en la opción **"Check for updates to CodeWarrior"** .....

5) Se desplegará una pantalla de actualización y procederemos a hacer click en **"next"** para avanzar en el proceso, según se muestra en la figura.



6) En la siguiente pantalla nos mostrará todos las actualizaciones disponibles y deberemos elegir todas (tildando las casillas) las referidas a los sistemas ***5.0*** y ***6.x*** para una correcta actualización.


CodeWarrior Updater
The following information was found about your products. Select which updates you wish receive. You can read a short description of the updates in the text box below.
CodeWarrior for HC08, Version 5.0
CW08 V5.0 LT8 Service Pack
CW08 V5.0 Patch 1
CW08 V5.0 LV8 Service Pack
CodeWarrior for HC08 V5.1
CW08 V5.1 GT16A Service Pack
CW08 V5.1 QY4A Service Pack
Download Updates Only: Do Not Install
Description
This is your currently installed version of the product. There are 11 updates found for this product.
freescale
Settings...
< Back
Next >
Cancel

A partir de esta pantalla se irán mostrando los progresos de las descargas y las posteriores instalaciones una a una a lo largo de todo el proceso de actualización.

Si se produjera algún error durante la instalación de alguno de los paquetes de actualización, el sistema lo corregirá en forma automática en posteriores intentos.

Se sugiere actualizar a los sistemas ***CodeWarrior*** en forma periódica.

Solo el sistema ***CodeWarrior 5.0*** **no posee** el **"service pack"** (patch) para manejar en forma transparente al sistema ***R(S)_POD*** que está basado en el OSBDM. Todas las demás versiones posteriores ya tienen incorportadas el "service pack" por lo que no nos deberemos preocupar por la compatibilidad del sistema ***R(S)_POD*** con los sistemas CodeWarrior.

4.2 - Instalación de los Drivers USB para el sistema *R(S)_POD.*

Como todo dispositivo USB, nuestro sistema ***R(S)_POD*** necesita una serie de Drivers para que el sistema ***Windows*** lo reconozca como un nuevo dispositivo USB conectado a un puerto de la misma.

El entorno de desarrollo integrado ***CodeWarrior*** emplea un archivo interface denominado **GDI** (**G**eneric **D**ebug **I**nstrument) que permite conectar una gran variedad de herramientas de desarrollo bajo un ambiente universal. El archivo GDI necesario para controlar la placa ***R(S)_POD*** conectada al puerto USB es:

- **"OpenSourceBDM_gdi.dll"** → Para el **CW 5.0.** **(solo para esta versión)**
- **"osbdm_s08.dll"** → Para el **CW 6.x** (familia HC9S08).
- **"osbdm_r08.dll"** → Para el **CW 6.x** (familia RS08).
- **"osdbm_CFV1.dll"** → Para el **CW6.x** (familia ColdFire V1).

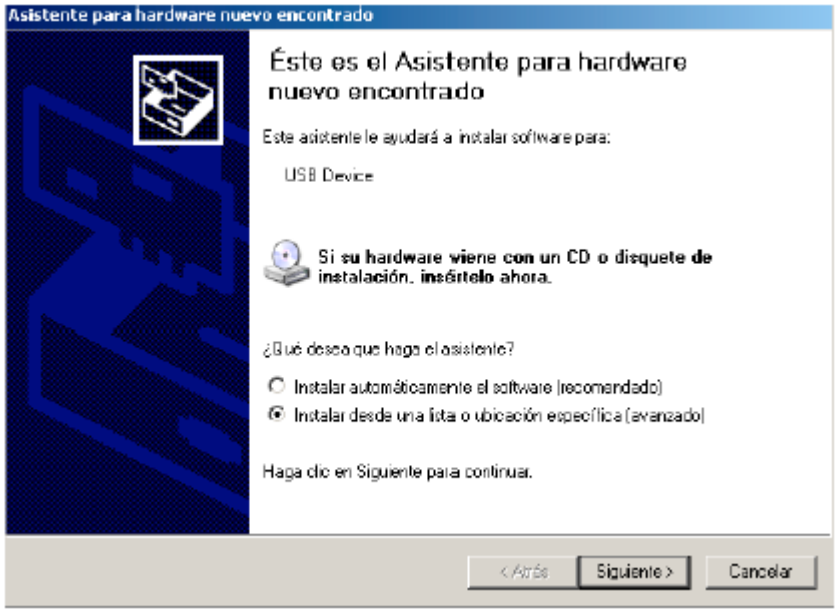
Los Drivers USB necesarios son:

- Open Source BDM Interface DLL.
- LIBUSB.LIB

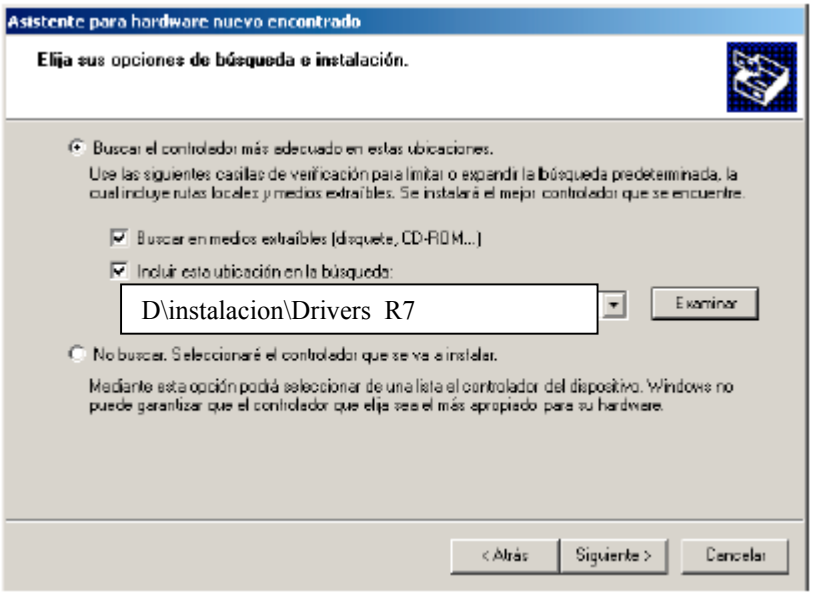
Proceder a instalar los Drivers USB del hardware según los siguientes pasos:

Nota: Los pasos aquí detallados corresponden al sistema operativo ***Windows XP*** *(para la versión de* ***Firmware R7*** *se recomienda* ***SP2*** *o* ***SP3, no usar SP1!!****)*, ya que es el sistema nativo de funcionamiento del entorno ***CodeWarrior.*** Para la instalación de los Drivers en un sistema **Windows 98SE** ver sección específica.

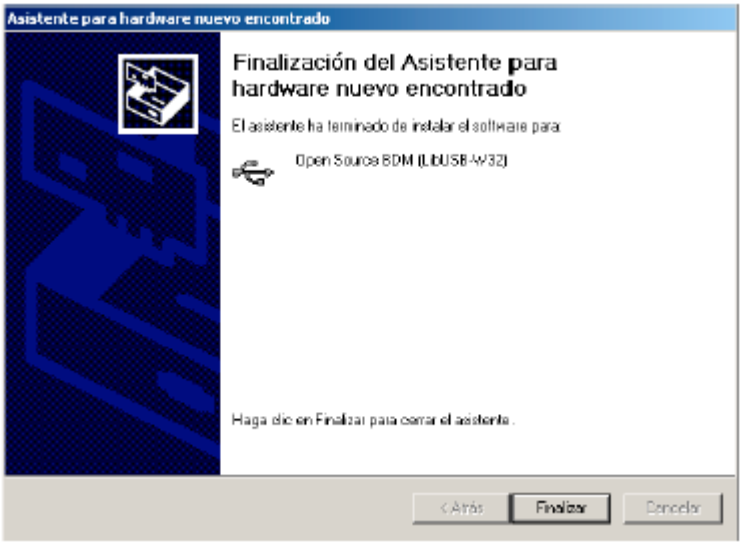
1) Conectar la placa ***R(S)_POD*** al puerto USB de la PC o Notebook por medio del cable USB serial tipo A – B provisto con el kit.


Asistente para hardware nuevo encontrado
Éste es el Asistente para hardware nuevo encontrado
Este asistente le ayudará a instalar software para:
USB Device
Si su hardware viene con un CD o disquete de instalación, insértelo ahora.
¿Qué desea que haga el asistente?
Instalar automáticamente el software (recomendado)
Instalar desde una lista o ubicación específica (avanzado)
Haga clic en Siguiente para continuar.
< Atrás
Siguiente >
Cancelar

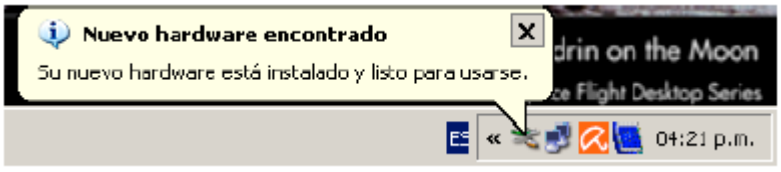
2) Al conectar la placa, ***Windows XP*** detectará la presencia de un nuevo dispositivo USB y nos mostrará una pantalla de inicio de instalación del nuevo hardware. Introducir el CD ROM de cortesía (instalación) en la PC, elegir la opción **"Instalar desde una lista o ubicación específica (avanzada)"** y hacer "click" en el botón **"Siguiente"**...


Asistente para hardware nuevo encontrado
Elija sus opciones de búsqueda e instalación.
Buscar el controlador más adecuado en estas ubicaciones.
Use las siguientes casillas de verificación para limitar o expandir la búsqueda predeterminada, la cual incluye rutas locales y medios extraíbles. Se instalará el mejor controlador que se encuentre.
Buscar en medios extraíbles (disquete, CD-ROM...)
Incluir esta ubicación en la búsqueda:
D\instalacion\Drivers R7
Examinar
No buscar. Seleccionaré el controlador que se va a instalar.
Mediante esta opción podrá seleccionar de una lista el controlador del dispositivo. Windows no puede garantizar que el controlador que elija sea el más apropiado para su hardware.
< Atrás
Siguiente >
Cancelar

3) En la siguiente pantalla, elegir la opción **"Buscar el controlador más adecuado en estas ubicaciones"** tildar los cuadros que se ven en la figura y ubicar la ruta de acceso a los DRIVERS USB en el CD ROM (como se puede observar en la figura) **(D:\instalacion\Drivers_R7)** y luego hacer click en el botón **"siguiente"**.....


Asistente para hardware nuevo encontrado
Finalización del Asistente para hardware nuevo encontrado
El asistente ha terminado de instalar el software para:
Open Source BDM (LibUSB-W32)
Haga clic en Finalizar para cerrar el asistente.
< Atrás
Finalizar
Cancelar

4) Por último, al finalizar el proceso de instalación, se abrirá una pantalla final que nos mostrará el software instalado (en este caso **"Open Source BDM (LibUSB-W32)**) y hacer click en **"Finalizar"** para salir del asistente de instalación.



4.3 - Instalación de los archivos GDI para la placa R(S)_POD.

Solo el entorno ***CodeWarrior 5.0*** necesita la instalación en forma manual de un archivo **GDI** (del tipo Open Source BDM Interface DLL) para poder garantizar la interface entre la PC – Drivers USB y el sistema ***CodeWarrior.*** El entorno ***CodeWarrior 6.x*** ya dispone de un "service pack" especial para instalar en **forma automática** estos archivos y otros más que garantizan el manejo transparente del sistema ***R(S)_POD***.

A continuación, se detallan los pasos para la instalación de los mismos **(solo para el sistema *CodeWarrior 5.0*)**:

1) Buscar en el CD ROM la carpeta **"Soporte CW5.0"** y copiar los siguientes archivos en donde se hayan instalado los archivos principales del sistema ***CodeWarrior 5.0*** (por ejemplo, C:\Archivos de programa\Freescale\CW08 V5.0\prog\gdi), más precisamente en la carpeta "GDI").

Para el CodeWarrior 5.0 copiar:

- **Opensourcedbm_gdi_s08_v3.dll**

2) Adicionalmente en el sistema ***CodeWarrior 5.0*** se debe instalar el archivo **"hcs08_common.fpp"** que está contenido en la carpeta **"FPP_update"** en el mismo directorio de "Soporte CW5.0", copiándolo en la carpeta en donde se hayan instalado los archivos principales del sistema ***CodeWarrior 5.0*** (Por ejemplo, C:\Archivos de programa\Freescale\CW08 V5.0\prog\fpp), más precisamente en la carpeta "FPP").

En la carpeta **"FPP"** ya existe una versión previa del **"hcs08_common.fpp"**, sobre-escribir el mismo con la versión contenida en el CD ROM.

5.0 - Instalación del Hardware, placa R(S)_POD.

Una vez finalizada la tarea de instalación del software, que comprende la instalación del entorno ***CodeWarrior*** y los drivers USB, necesitamos conocer los alcances de los distintos "jumpers" presentes en la placa principal del sistema ***R(S)_POD.***

Si bien el sistema no necesita instalación alguna de hardware, los distintos jumpers disponibles permiten configurar al sistema en distintos modos de operación otorgándole mayor flexibilidad de uso.

El kit del sistema ***R(S)_POD*** está constituido por dos placas:

- **Placa principal "*R(S)_POD*"** --- Es la herramienta de desarrollo propiamente dicha.

- **Placa "*DemoQG8*"** ---- Placa de Demostración y práctica con el MCU **MC9S08QG8CPE** de la familia HC9S08.

El agregado de la placa **"*DemoQG8*"** le permite al usuario una rápida adaptación a la familia **HC9S08** y la posibilidad de efectuar sus primeros pasos de una forma amena y sencilla.

Para comenzar a trabajar con ambas placas, debemos conectarlas entre sí y a la PC o Notebook según se muestra en la **figura 1.**

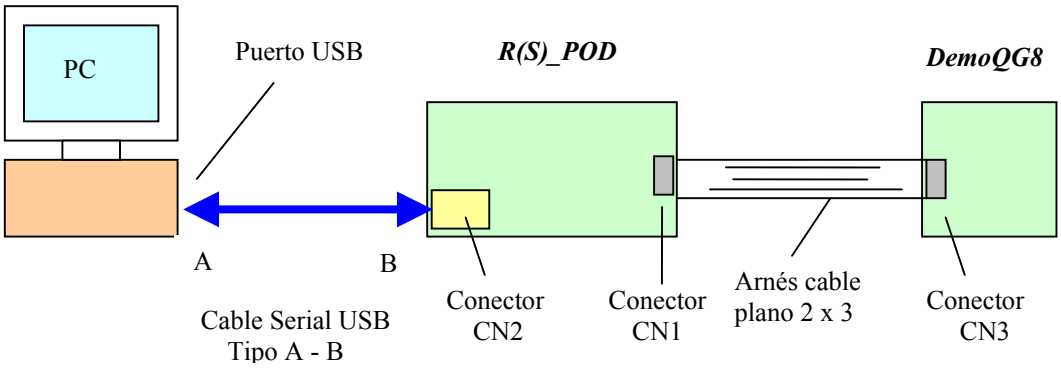


Figura 1.- Conexión entre placas y PC

6.0 – Configuración de Jumpers / Switchs & LEDs.

La placa principal del sistema ***R(S)_POD*** posee una serie de "jumpers", Switchs y LEDs indicadores que permiten configurar al sistema en distintos modos de operación que le otorgan una gran flexibilidad de uso y claras indicaciones de funcionamiento.

Jumpers:

JP1 ----- "Power_Select" ----- Selección de Alimentación "VDD".

A) Posición 1-2 → Interface BDM alimentada por la aplicación (tarjeta usuario).

B) Posición 2-3 → El sistema R(S)_POD le provee alimentación a la aplicación.

A) En la posición 1-2, los circuitos de la interface BDM (entre la herramienta y la placa de aplicación del usuario) en el sistema R(S)_POD son alimentados por la propia placa del usuario por medio del "pin 6" (VDD) del conector BDM (ver sección 7.0 "Conector BDM").
Cuando se elija esta opción, se debe tener en cuenta que **la placa del usuari**o (placa bajo desarrollo) **es la que debe proveer de alimentación** a los circuitos de la interface BDM, estando comprendida entre +1,8V a +5V como máximo, con un consumo aproximado de 1mA de corriente. Todas las demás funciones del sistema R(S)_POD se alimentan por medio del puerto USB de la PC o Notebook utilizados.

La opción **"Interface BDM alimentada por la aplicación"** es muy útil cuando en la placa bajo desarrollo se dispone de una fuente de alimentación determinada o razones especiales de consumo (Por ejemplo, corrientes superiores a los 300 mA) que hacen impracticable el suministro de energía por otros medios. También resulta útil en aplicaciones que deben depurarse "en campo" sin la posibilidad de alimentarse en forma separada.

B) En la posición 2-3, **el sistema R(S)_POD le provee de alimentación a toda la placa del usuario bajo desarrollo,** esto significa que la placa del usuario NO NECESITA de alimentación externa alguna, ya que ahora el "pin 6" (VDD) del conector BDM (ver sección 7.0 "Conector BDM"), le provee alimentación a la aplicación del usuario.
La tensión disponible en dicho pin (que es la VDD del MCU bajo desarrollo) es AJUSTABLE por medio de preset PR1 entre los valores de +1,8V a +5V con una corriente máxima de 300 mA.
Al igual que en la opción anterior, el sistema ***R(S)_POD*** obtiene su alimentación desde el puerto serial USB de la PC o Notebook utilizado.

La opción **"R(S)_POD provee alimentación a la aplicación"** es útil cuando se está en plena tarea de desarrollo y no se quiera utilizar o implementar circuitos de alimentación o fuentes externas que dificulten la portabilidad de la aplicación bajo desarrollo.

Nota Importante:

Como la herramienta ***R(S)_POD*** obtiene su alimentación a través de un puerto serial USB, el usuario debe verificar que el mismo es del tipo **"HOST" versión USB 2.0** ya que el mismo por norma puede entregar una corriente **MÁXIMA de 500 mA.**
Si bien la herramienta cuenta con un sistema de protección y aviso automático por sobre – corriente, este mecanismo **NO asegura la integridad del puerto serial USB.**
Se sugiere al usuario, tener especial cuidado con las corrientes máximas entregadas por la herramienta, así como las diferencias de potencial producidas por deficientes puestas a tierra de la PC, retornos de corriente de la placa bajo desarrollo, o falta de aislación galvánica entre los distintos sistemas, **ya que podrían provocar daños graves en el puerto serial de la PC o en el funcionamiento de la herramienta.**

JP2 ---- Indicación "MCU_POWER" ---- Habilitación / Deshabilitación LD3.

JP2 → Cerrado → LD3 Habilitado → Indicación "MCU_POWER" Habilitada.

JP2 → Abierto → LD3 Deshabilitado → Indicación Deshabilitada.

Con el jumper **JP2** se habilita / deshabilita la indicación de **"MCU_POWER"** por medio del **led "LD3".** El encendido de LD3 corresponde a la presencia o no de alimentación en el pin VDD del MCU bajo desarrollo. El sistema ***R(S)_POD*** dispone en su conector BDM el pin 6 de VDD que está vinculado a la tensión de alimentación del MCU bajo desarrollo, por lo que el encendido o no del LED o su variación de intensidad de iluminación corresponderá a los valores de VDD.
Como LD3 consume energía de VDD el usuario puede juzgar cuando habilitar o deshabilitar la indicación según sus condiciones de desarrollo.

JP3 / JP4 / JP5 --- Habilitación / Deshabilitación de actualización por ICP.

JP3 → Posición 1-2 → Posición normal de trabajo (Posición de Fábrica).

JP3 → Posición 2-3 → Sistema Habilitado p/ actualización de firmware por ICP.

JP4 / JP5 → ABIERTOS (sin gota de estaño) → Estado normal (Fábrica).

Los **jumpers JP3 / JP4 / JP5** (JP4 / JP5 son jumpers del tipo "gota de estaño") permiten que el sistema ***R(S)_POD*** pueda ser actualizado (actualización de firmware) en circuito (ICP) por medio del **puerto Serial USB** y un aplicativo en la PC especialmente diseñado para tal fin por Freescale Semiconductor.

Durante el uso normal de la herramienta estos jumpers deben permanecer en las posiciones originales de fábrica, para evitar que por "accidente" se corrompa el firmware original de fábrica contenida en la misma.

Se sugiere ver la sección **"Actualización de Firmware"** en el presente manual.

Switch SW1 con retención:

SW1 --- "POWER MCU ON/OFF" ---- Control de la alimentación VDD al MCU.

SW1 → PULSADO → VDD ON → VDD presente en pin 6 conector BDM (CN1).

SW1 → LIBERADO → VDD OFF → VDD NO disponible en pin 6 de CN1.

Nota 1: Si el jumper **"JP1"** está en la **posición 1-2** (interface BDM alimentada por la Aplicación), el switch **SW1 no tiene acción alguna en el control de VDD**, ya que la misma es provista por la misma aplicación.

Nota 2: En la familia **HC9S08 / ColdFire V1** no es necesario tener control de la alimentación VDD tanto para la grabación de la memoria FLASH como cuando se está en pleno proceso de depuración de programa. Por lo tanto, siempre se debe dejar la VDD presente en el circuito a desarrollar.
En la nueva familia **RS08** si es necesario tener control de la alimentación VDD durante el proceso de grabación de la memoria FLASH. Para mayor información, sugerimos Consultar el manual de datos del dispositivo RS08 utilizado.

LEDS Indicadores:

LD1 ---- "VPP_EN" ---- Tensión VPP ON / OFF (Solo para familia RS08).

LD1 → Encendido → VPP ON → VPP (+12V) presente en pin RST / VPP del conector BDM (CN1).

LD1 → Apagado → VPP OFF → Solo presente la función RESET en pìn RST/VPP del conector BDM (CN1).

La familia **RS08** necesita una tensión externa elevada llamada **"VPP"** durante el proceso de grabación o borrado de la memoria FLASH, ya que esta familia **NO incorpora la "bomba de carga"** que si tienen en forma interna las familias **HC908** y **HC9S08 / CFV1.** Esta tensión es para los modelos actuales de **+12V** y el sistema ***R(S)_POD*** ha sido diseñado para proveerla en forma totalmente automática, ya que cuando se trabaja con la familia **HC9S08** esta tensión no está presente en el **"pin 4"** del **conector BDM (CN1)** que solo tiene la función de **RESET** del MCU bajo desarrollo.

Nota: La tensión VPP viene ajustada de fábrica a un valor de +12V, la misma puede ajustarse por medio del preset **PR2 "VPP_ADJ"** que incorpora la placa de la herramienta. Para medir VPP debe colocarse un voltímetro entre los pines + y (-) del capacitor electrolítico C14. Variaciones de **+/- 0,3V** son perfectamente aceptables.

LD2 --- "USB_EN" ---- Actividad en el puerto USB entre R(S)_POD y PC.

LD2 → Apagado → Sistema R(S)_POD no conectado a la PC o puerto USB con problemas (R(S)_POD sin alimentación).

LD2 → Encendido fijo → Sistema alimentado / Ready → No hay actividad.

LD2 → Encendido destellante → Actividad del puerto USB (ACTIVO).

El LED **LD2 "USB_EN"**, es útil para monitorear la actividad de la comunicación en el puerto USB entre el sistema ***R(S)_POD*** y la PC. Además permite verificar que la herramienta está perfectamente alimentada, ya que, si el LED se encontrara APAGADO y la herramienta perfectamente conectada al puerto USB de la PC, ello podría significar que el puerto no funcionaría correctamente o el cable Serial USB no se encontrara bien conectado o no funcionaría correctamente.

LD3 --- "MCU_POWER" ----- Indicación de VDD ON / OFF en el MCU.

LD3 → Apagado → VDD OFF → no hay alimentación presente en "pin 6" (VDD) del conector BDM (CN1) y en el MCU.

LD3 → Encendido → VDD ON → hay alimentación presente en "pin 6" (VDD) del conector BDM (CN1) y en el MCU.

El **LED "LD3"** sirve para "monitorear" la presencia o no de la alimentación VDD del MCU bajo desarrollo. Según el esquema de operación elegido por medio del jumper **"JP1"** tendremos o no VDD en forma permanente.

LD4 ---- "OverCurrent" ---- Indicación de "Sobre-Corriente" en VDD.

LD4 → Apagado → Estado NORMAL de la corriente en VDD (corriente entregada a la placa bajo desarrollo).

LD4 → Encendido → Sobre – Corriente en VDD (la corriente entregada a la placa es mayor a lo aconsejable).

El sistema ***R(S)_POD*** dispone de un mecanismo automático de protección contra sobre-corriente en la alimentación de VDD cuando esta es entregada por la herramienta. De esta forma, el circuito del usuario (placa bajo desarrollo) se encuentra protegido ante excesos de corriente en la alimentación del MCU (VDD) y del resto del circuito. Además en forma indirecta, y gracias al mismo mecanismo, se encuentra también protegido el puerto serial USB de la PC o Notebook conectada a la herramienta.

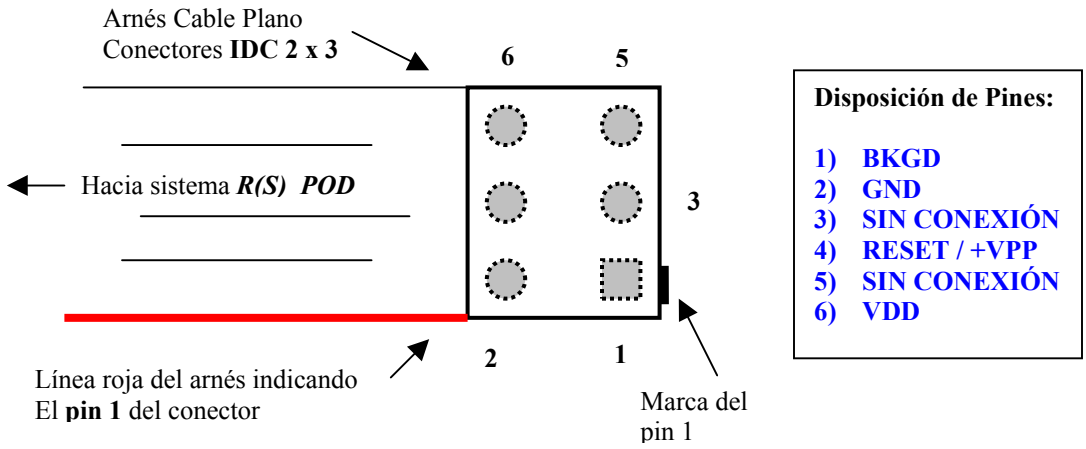
Nota: Como todo circuito de protección no es 100% seguro, se sugiere al usuario extremar los cuidados durante el uso del sistema para evitar daños graves tanto en el puerto serial USB de la PC / Notebook como en la propia herramienta y/o placa bajo desarrollo.

7.0 - Conector BDM para la conexión con la tarjeta de usuario.

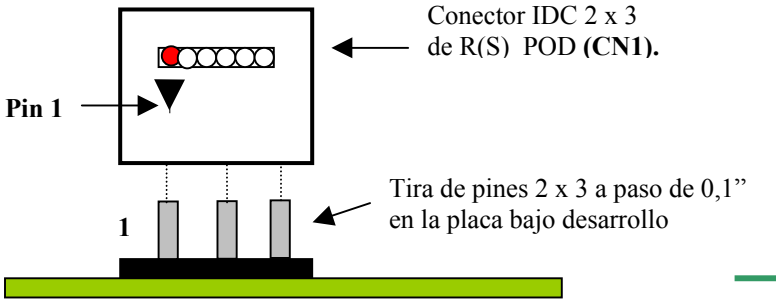
Freescale Semiconductor ha adoptado una configuración de pines reunidos en un conector para vincular el sistema bajo desarrollo del usuario (aplicación) y las distintas herramientas disponibles para las familias FLASH **HC9S08 / RS08 / HC9S12 y Serie Flexis 8/32 Bits.**

Esta disposición estandar de pines se conoce como **"BDM Connector" (Conector BDM)** y consiste en un **conector del tipo IDC 2 x 3 pines ("Berg")** en un extremo del arnés de cable plano que **se conecta con la placa del usuario por medio de una tira de 2 x 3 pines a paso de 0,1" (pulgadas) (6 pines en total).** De esta forma, toda aplicación que utilice algún miembro de la familia FLASH HC9S08 / RS08 / HC9S12 / Serie Flexis HC9S08 / ColdFire V1, podrá hacer uso de una gama muy amplia de herramientas de Freescale y de terceras partes **sin problemas de compatibilidad en la conexión entre placa – herramienta.**

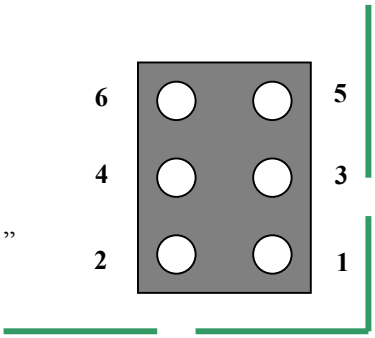
El sistema ***R(S)_POD*** utiliza este estandar **(conector CN1)** y la configuración de pines es la siguiente:



Arnés + conector BDM (CN1) visto desde arriba



Detalle de conexión entre CN1 – Tira de pines de la placa bajo desarrollo.



Tira de pines 2 x 3 a paso de 0,1" Dispuesta en la placa bajo desarrollo

Descripción de Pines:

Pin 1 → BKGD → "Background Debug" → Pin de comunicación a un solo hilo utilizado durante el modo "Background Activo" o durante el envío de comandos no intrusivos desde la herramienta BDM (R(S)_POD) y el MCU bajo desarrollo. Debe conectarse al pin BKGD del MCU bajo desarrollo en la placa del usuario.

Pin 2 → GND → Ground (VSS) → Pin de masa del sistema, debe conectarse a GND (VSS) de la placa del usuario.

Pin 3 → N.C. → Sin Conexión → Dejar libre en el PCB de la aplicación.

Pin 4 → RESET / +VPP → Señal de RESET activa baja y de +VPP → Este pin controla la señal RESET de los MCUs HC9S08 cuando así se lo requiera y/o la tensión de borrado / programación de la memoria FLASH "+VPP" de la familia RS08. Debe conectarse al pin de RESET del MCU cuando así se lo disponga (existen algunos derivativos de la familia HC9S08 que no disponen del pin de RESET). El uso de la señal de RESET garantiza mayor control del dispositivo ante situaciones de no control por medio del pin BKGD.

Pin 5 → N.C. → Sin Conexión → Dejar libre en el PCB de la aplicación.

Pin 6 → VDD → Alimentación VDD del MCU bajo desarrollo → Según el modo de operación elegido por medio del Jumper "JP1" el sistema ***R(S)_POD*** **"recibirá"** una tensión VDD desde el MCU bajo desarrollo (placa del usuario) alimentando al circuito de interface BDM o bien **"entregará"** una tensión VDD a la placa bajo desarrollo alimentando de esta forma a la misma.
Debe conectarse a +VDD del MCU y placa bajo desarrollo.

Nota 1: Tener cuidado de respetar las tensiones máximas de trabajo para cada familia de microcontroladores utilizada, fundamentalmente cuando el sistema ***R(S)_POD*** provea de alimentación a la placa bajo desarrollo.

Nota 2: Se sugiere consultar el manual de datos del microcontrolador utilizado para conocer en forma detallada el funcionamiento de cada una de las señales involucradas con el sistema BDM, para de esta forma obtener el mayor provecho del sistema de desarrollo.

8.0 – Poniendo en marcha el Sistema.

Para aprender a usar el sistema ***R(S)_POD*** nada mejor que "poner mano" en una aplicación de ejemplo. Para ello utilizaremos la placa ***DemoQG8*** contenida en el kit y que dispone de un microcontrolador MC9S08QG8CPE de 16 pines en encapsulado DIP con una buena cantidad libre de ellos para nuestra aplicación.

El programa de ejemplo se basará en el uso del módulo de Timer que dispone esta familia para generar una interrupción por "Timer Overflow" (Time Out) cada 250 ms que dará lugar al disparo de una serie de tareas (en este caso solo la inversión de estado de un pin de un puerto I/O) como lo haría un sistema RTOS (Real Time Operate System) o también conocido como programa "Calesita". Se utilizará el entorno ***CodeWarrior 6.2***, ya que como veremos, posee mayores opciones de control del sistema ***R(S)_POD*** gracias al **"OSBDM Service Pack"** incorporado en este.

El ejemplo, paso a paso:

1) Configurar el hardware de la siguiente manera.....

- **Jumpers / Switch SW1:**

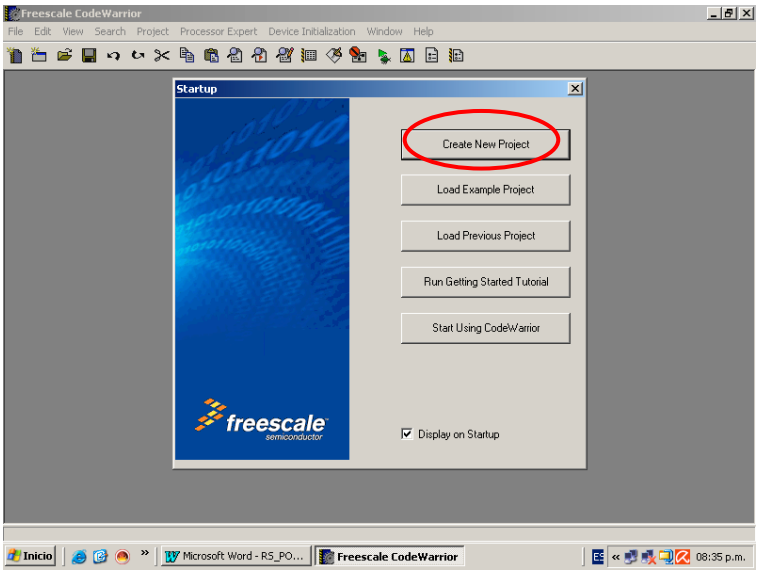
  JP1 → Posición 2-3.
  JP2 → Posición "Cerrado".
  JP3 → Posición 1-2 (de fábrica – No tocar!).
  JP4 / JP5 → Abiertos (de fábrica – No tocar!).
  SW1 → No Pulsado (aplicación sin VDD).

- Conectar la placa ***R(S)_POD*** con la placa ***DemoQG8*** por medio del arnés de cable plano con conectores IDC de 2 x 3 (CN1 – CN3).

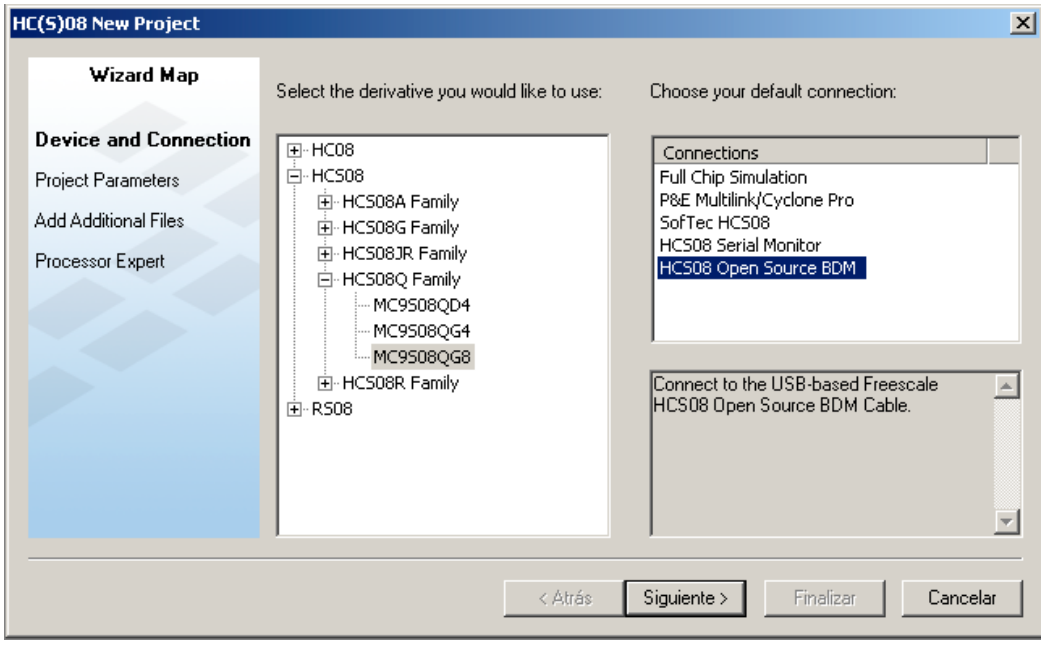
2) Conectar las placas así configuradas al puerto USB de la PC por medio del cable USB Tipo A-B provisto con el kit (en el sistema ***Windows XP*** debe escucharse un "Ding / Dong" característico de la detección del dispositivo USB y debe encenderse en forma "fija" el led verde LD2 "USB_EN" indicando sistema "Ready").

3) Abrir el entorno ***"CodeWarrior 6.2"*** por el ícono de acceso directo en el escritorio o bien por medio de la barra de inicio.

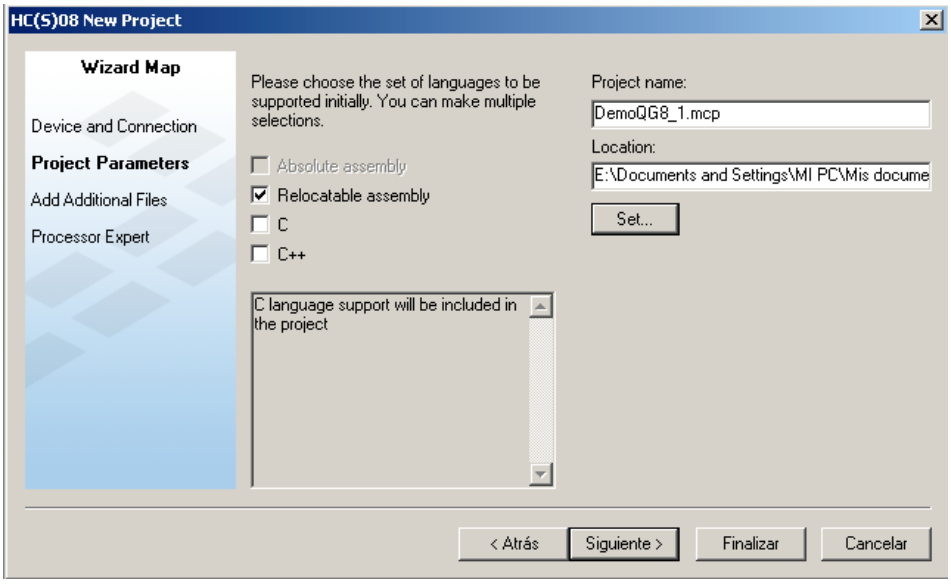
4) Elegir la opción "Create a New Project" según se muestra en la figura .....



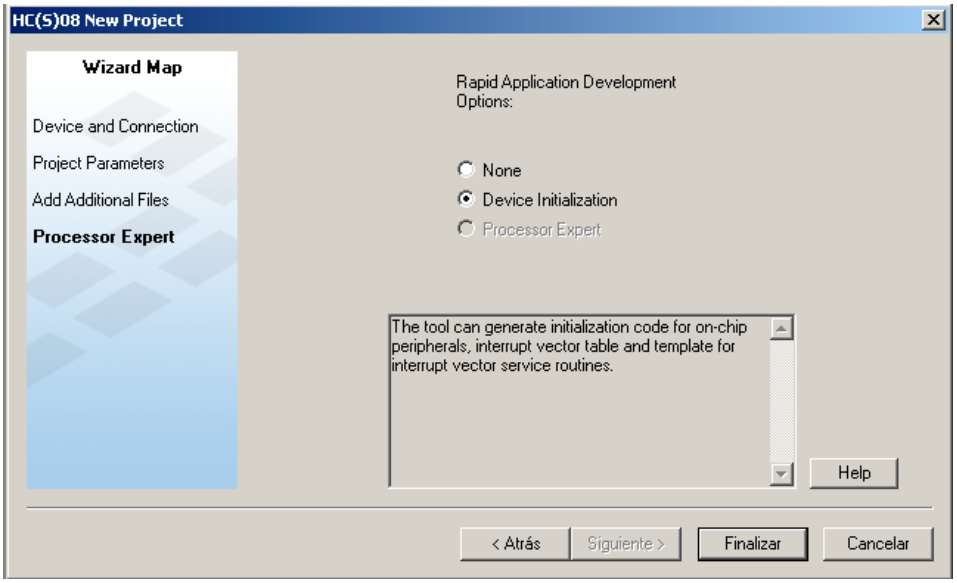
5) En la ventana **"Device and Connection"** elegir el dispositivo **"MC9S08QG8"** y la conexión **"HCS08 Open Source BDM"** que es la interface utilizada para comunicar el sistema ***R(S)_POD*** con el entorno CodeWarrior y hacer click en el botón "Siguiente".



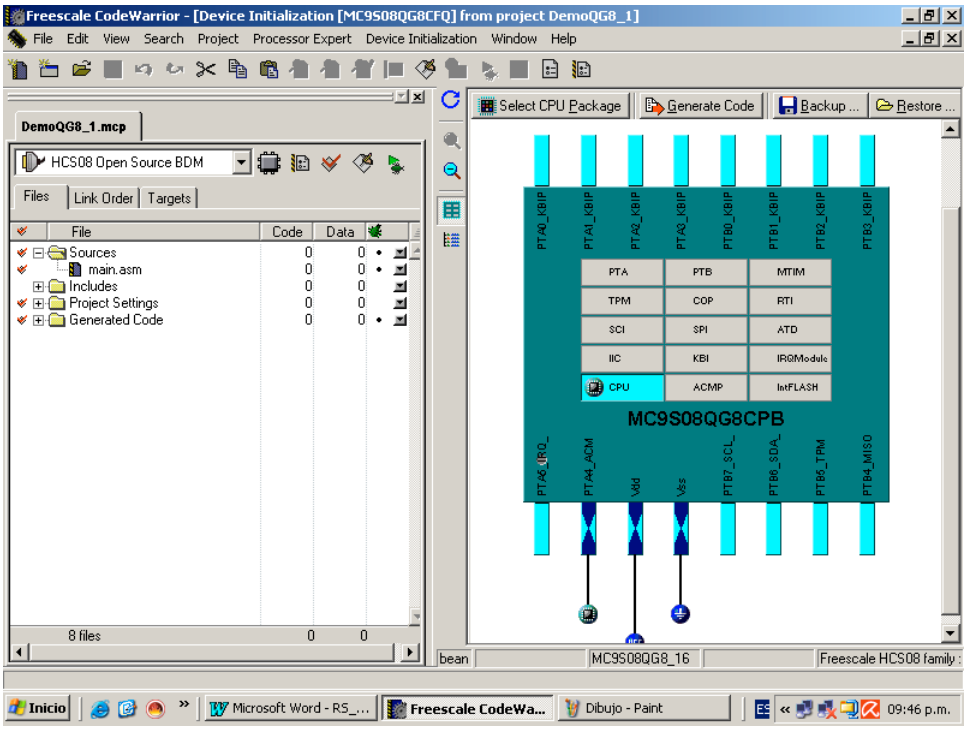
6) En la ventana **"Project Parameters"** elegir la opción **"Relocatable assembly"** y darle un nombre al proyecto como, por ejemplo, **"DemoQG8_1.mcp"**, y hacer click en siguiente....



7) En la ventana **"Processor Expert"** elegir la opción **"Device inicialization"**, así de esta forma la aplicación **"Processor Expert"** nos irá guiando en la configuración de los distintos registros del MCU elegido para el ejemplo.

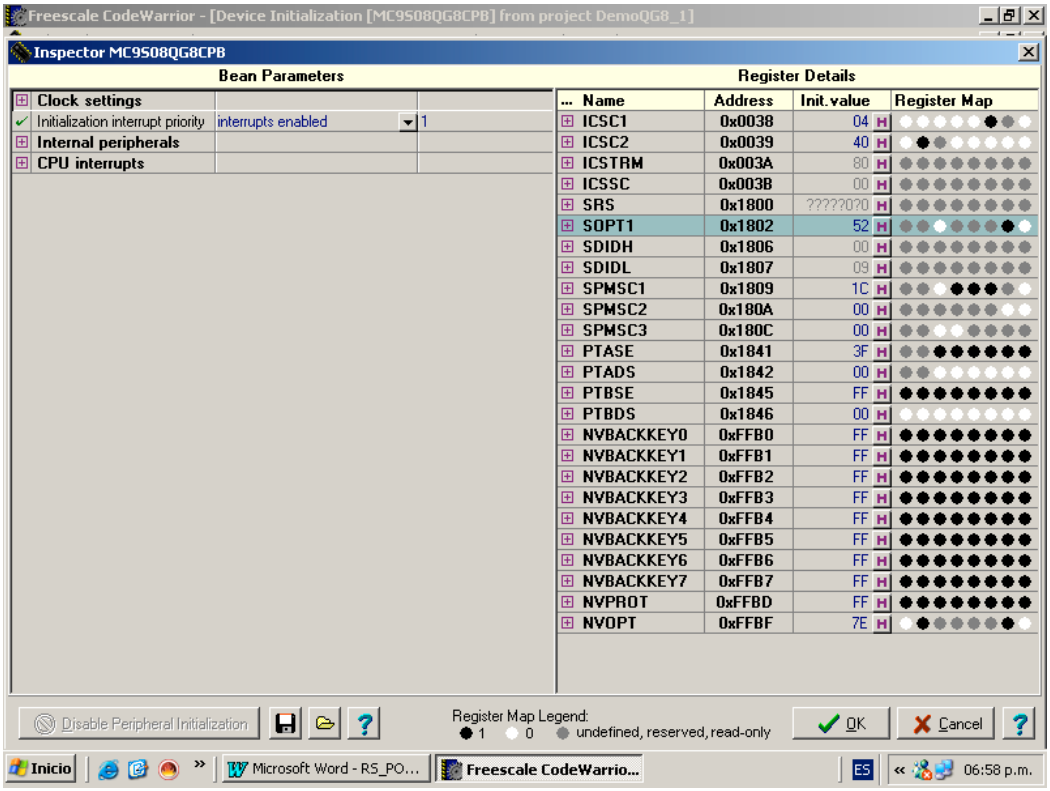


8) Hacer click en **"finalizar"** para "lanzar" el processor expert y comenzar a trabajar en el.



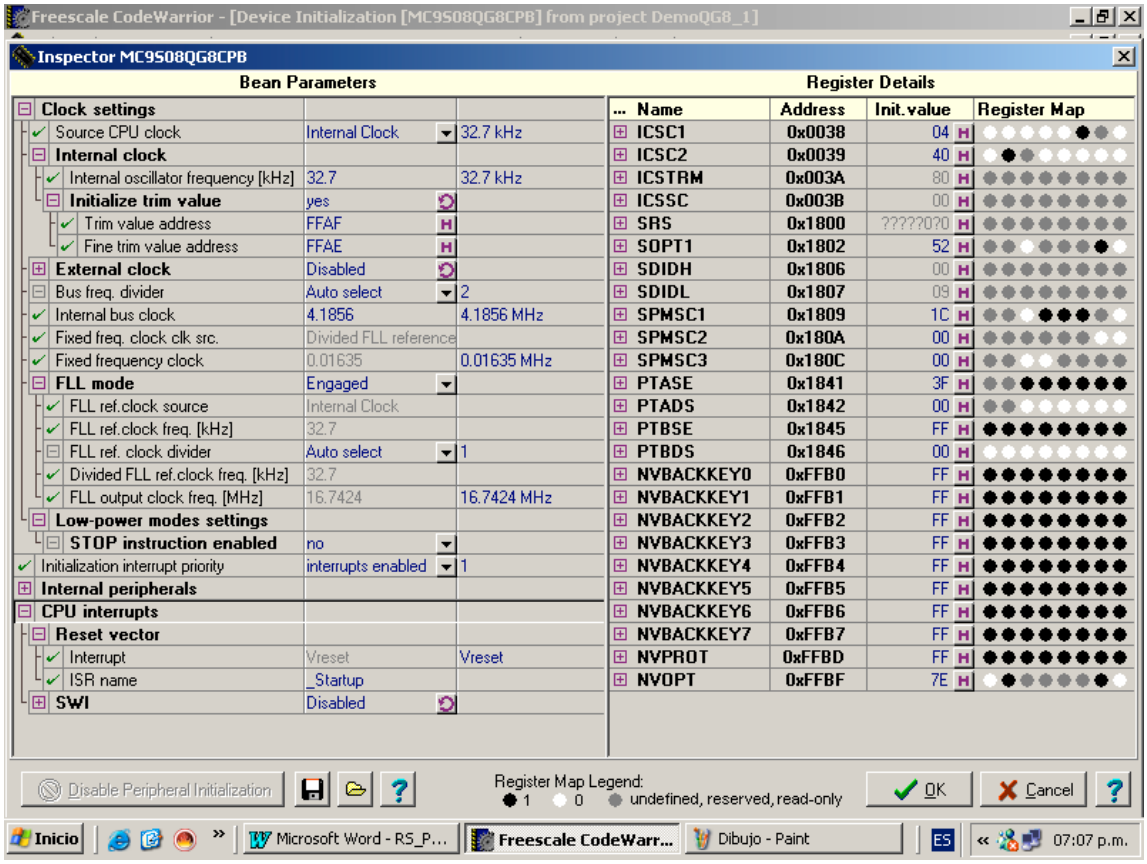
9) En la pantalla principal del proyecto, se podrán observar dos cuadros, uno conteniendo todas las carpetas del proyecto y otro con un entorno gráfico del chip bajo desarrollo. Elegir el encapsulado de **16 pines DIP** (MC9S08QG8CPB) haciendo click en el icono **"Select CPU Package".**

10) A continuación, se procederá a generar el código de inicialización del Timer para producir una interrupción periódica que será la base de nuestro sistema de disparo de tareas, inicializar los puertos I/O, los registros de configuración, etc., etc. Para hacer esto, usaremos el generador de código asistido "Processor Expert" haciendo click primeramente en el **módulo CPU** para configurar el clock del sistema y otros aspectos entorno del mismo.



Se configurará el módulo de CPU para:

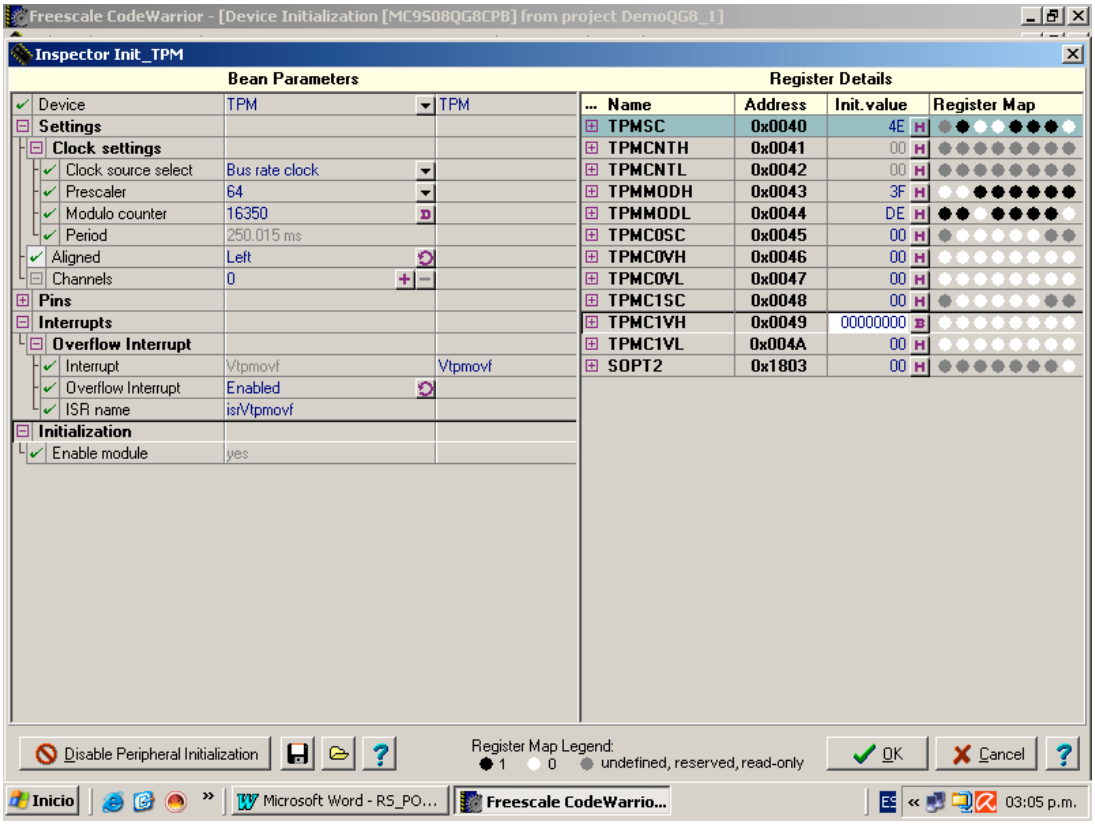
- Clock ---- Interno ---- 32,7Khz (referencia) --- FLL activo ---- 16,7424 Mhz
- Frecuencia de Bus = 4,1856 Mhz ---- Divisor de Bus = 2 (OSC / 4).
- LVD --- Habilitado ----- LVD deshabilitado en modo STOP.
- Interrupciones Habilitadas.
- Vector de Reset apuntando a la etiqueta "_Startup".
- Pin de Reset Externo no disponible.
- SWI deshabilitada.



11) Una vez configurados los distintos puntos del cuadro de configuración del módulo de CPU, hacer click en el icono **"O.K"** para confirmar los cambios y el sistema volverá a la pantalla principal. Ahora será el turno de la configuración del **módulo de TIMER** haciendo click sobre el cuadro con el nombre **"TPM".**

12) Se configurará el mismo con los siguientes parámetros:

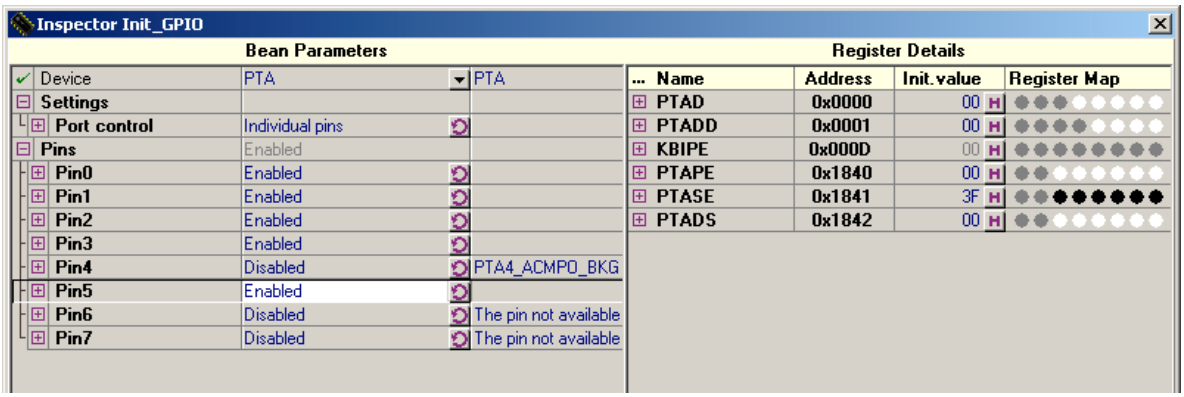
- Selección de fuente de Clock → Bus rate Clock.
- Prescaler → 64.
- Módulo Counter → 16350 cuentas.
- Período → 250 mSeg.
- Alineado → izquierda.
- Pines → No habilitados.
- Interrupción por "Overflow" → habilitada → **"isrVtpmovf" (etiqueta de la Interrupción).**
- Inicialización → Módulo habilitado.

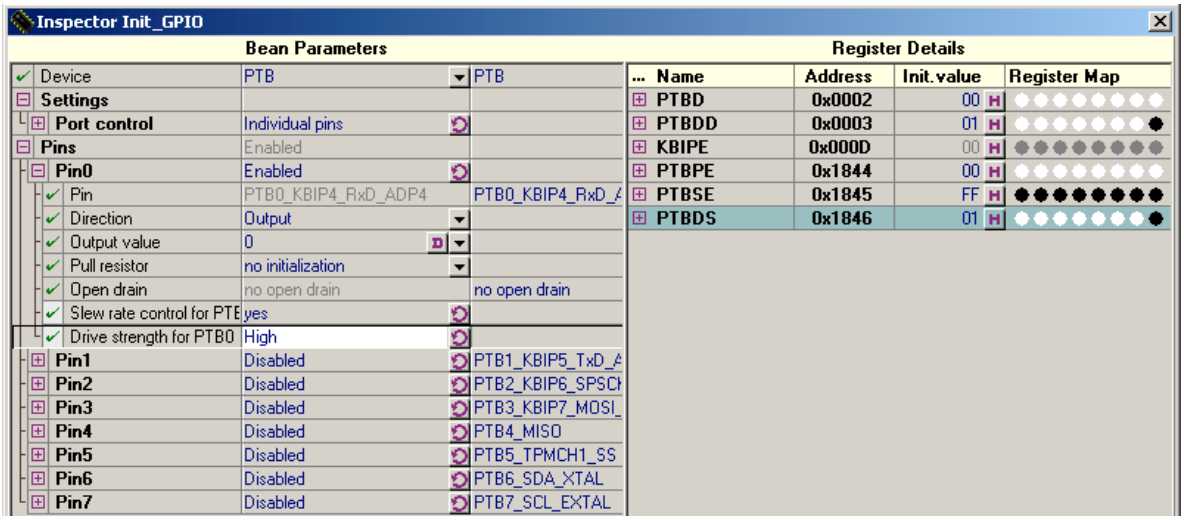


13) Al igual que en el módulo anterior, se procederá a hacer click en el botón ***"O.K"*** para confirmar la configuración de este y volver a la pantalla principal. Luego será el turno de los módulos de los puertos A y B (PTA / PTB) que los configuraremos de la siguiente manera:

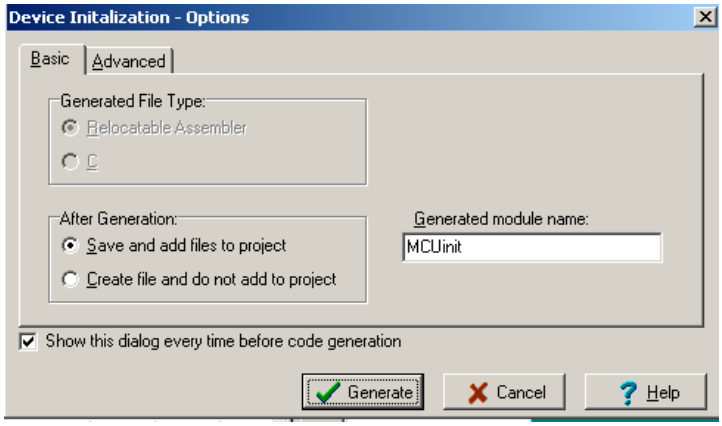
PTA (PORT A) → PTA0 / PTA3 → Habilitados → Input
→ PTA4 → No habilitado
→ PTA5 → Habilitado → Input
→ PTA6 / PTA7 → No Habilitados.

PTB (PORT B) → PTB0 → Habilitado → OUTPUT (control de corriente HIGH).
→ Valor de arranque → "0".
→ PTB1 / PTB7 → No Habilitados

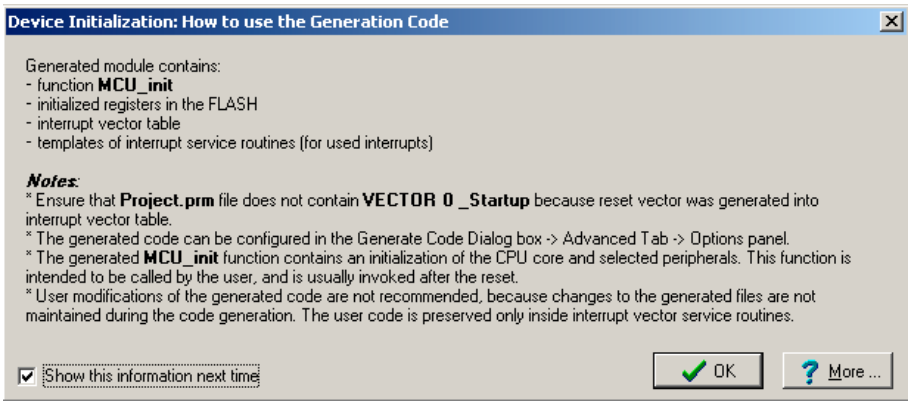




14) Si luego se presiona el botón ***"Generate Code",*** el generador de código del Processor Expert generará código y nos mostrará una ventana explicando los pasos a seguir para incorporarlo efectivamente al resto del programa.



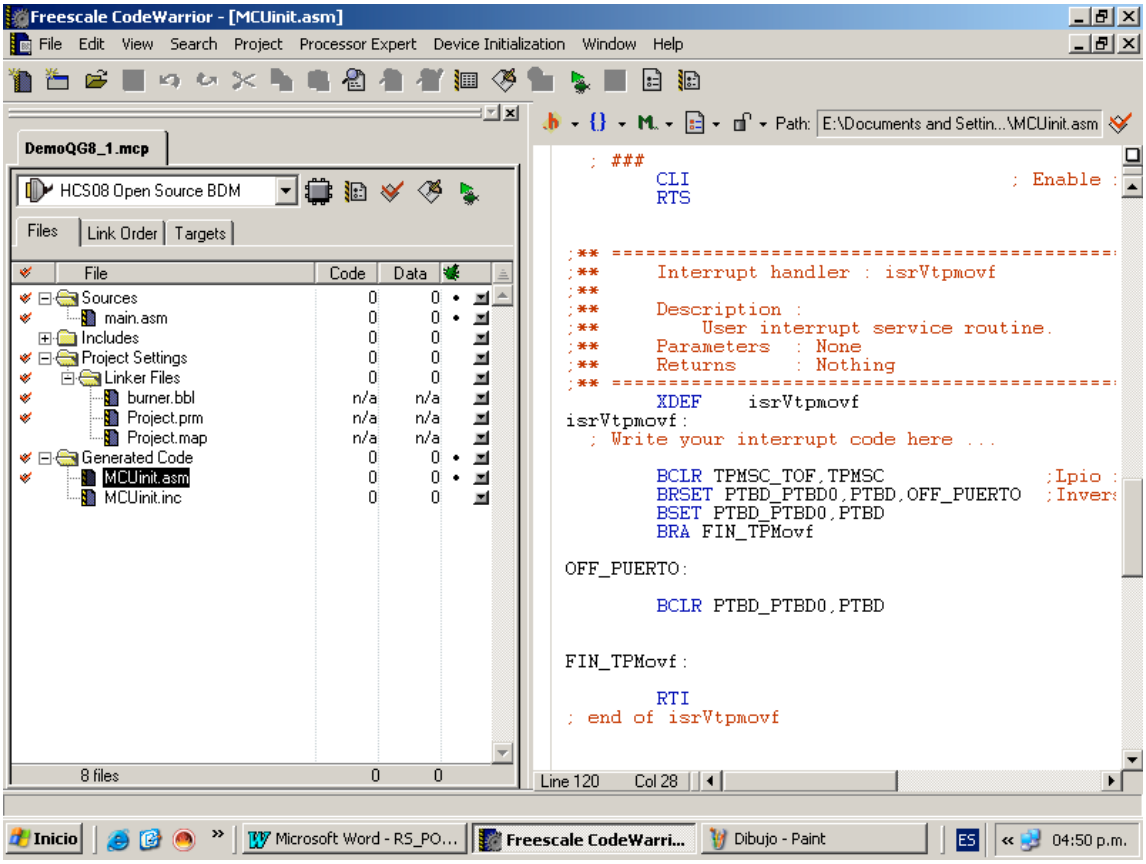
Se abrirá una pantalla previa que nos dará distintas alternativas de seteos (no tocar, solo para expertos!!) y nos dirá que el código generado llevará el nombre del módulo **"MCUint"** y se deberá proceder a presionar el botón **"Generate"** .....


Device Initialization: How to use the Generation Code
Generated module contains:
- function MCU_init
- initialized registers in the FLASH
- interrupt vector table
- templates of interrupt service routines (for used interrupts)
Notes:
* Ensure that Project.prm file does not contain VECTOR 0 _Startup because reset vector was generated into interrupt vector table.
* The generated code can be configured in the Generate Code Dialog box -> Advanced Tab -> Options panel.
* The generated MCU_init function contains an initialization of the CPU core and selected peripherals. This function is intended to be called by the user, and is usually invoked after the reset.
* User modifications of the generated code are not recommended, because changes to the generated files are not maintained during the code generation. The user code is preserved only inside interrupt vector service routines.
Show this information next time
OK
More ...

Otra pantalla nos dará algunas recomendaciones a tener en cuenta y se procederá a hacer click en el icono **"O.K"** para obtener finalmente el código generado.

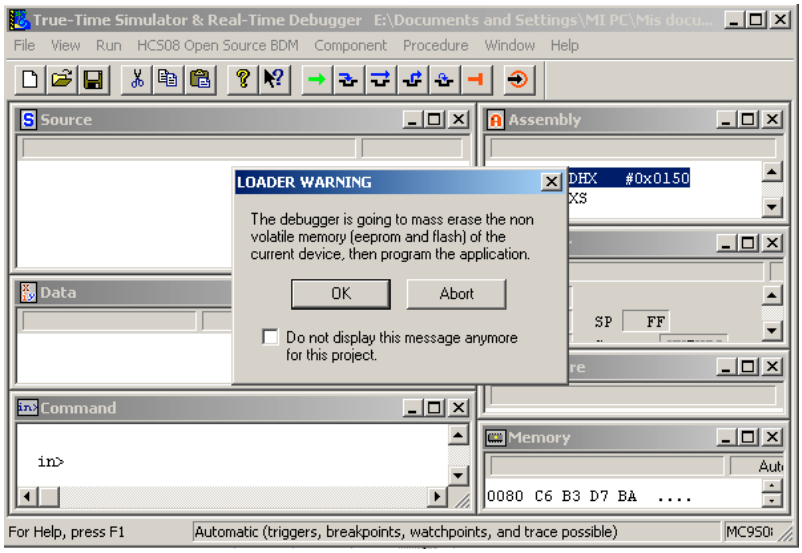
15) Hecho esto, se estará en condiciones de introducir nuestras líneas de código en la sub rutina de interrupción por Timer Overflow (**isrVtpmovf**) ubicada dentro del módulo generado MCUint (**MCUint.asm**) para realizar, por ejemplo, un Toggle (inversión de estado) cada vez que atendemos la interrupción propiamente dicha. En este punto podemos poner todas nuestras tareas en forma de llamado a sub rutina que se irán ejecutando una a una cada 250 ms. Las líneas a agregar pueden ser las siguientes...

```
     BCLR TPMSC_TOF,TPMSC                ;Lpio flag TOF del TIMER
    BRSET PTBD_PTBD0,PTBD,OFF_PUERTO     ;Inversión del puerto...
    BSET PTBD_PTBD0,PTBD
    BRA FIN_TPMovf

OFF_PUERTO:

    BCLR PTBD_PTBD0,PTBD


FIN_TPMovf:

    RTI
```



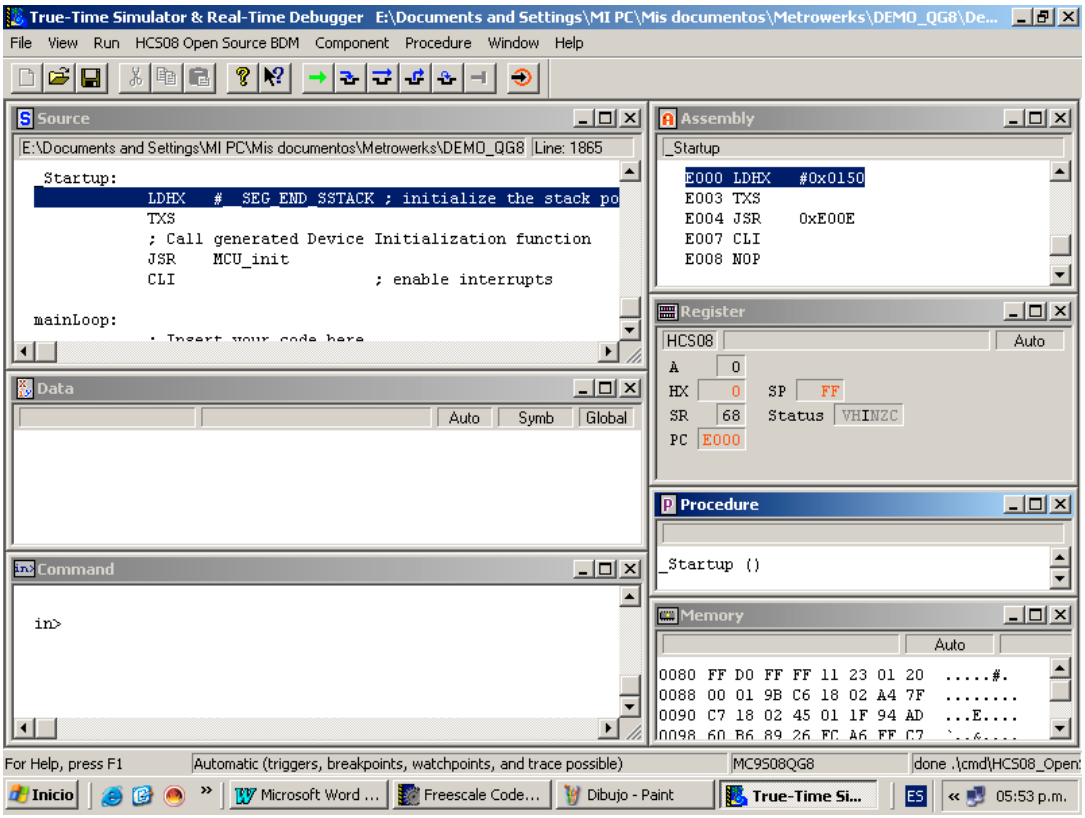
16) Una vez introducido nuestro código, se deberá compilar haciendo click en el botón **"Make"** en la barra de proyecto o en la barra de herramientas general. Si no se ha cometido algún error de compilación se estará en condiciones de pasar a la etapa de EMULACIÓN EN TIEMPO REAL del programa de ejemplo.

Para realizar ello, primero deberemos verificar que se haya elegido la opción **"HCS08 Open Source BDM"** en el cuadro de selección de herramientas y que se encuentre conectado el sistema ***R(S)_POD*** al puerto USB como se detalló en párrafos anteriores. Presionar el switch **SW1** con retención y deberá encenderse el **LED LD3** (**MCU_PWR**). Hacer click en la fecha **"verde"** (Debug) junto al cuadro de selección de herramientas como se puede ver en la siguiente figura.....

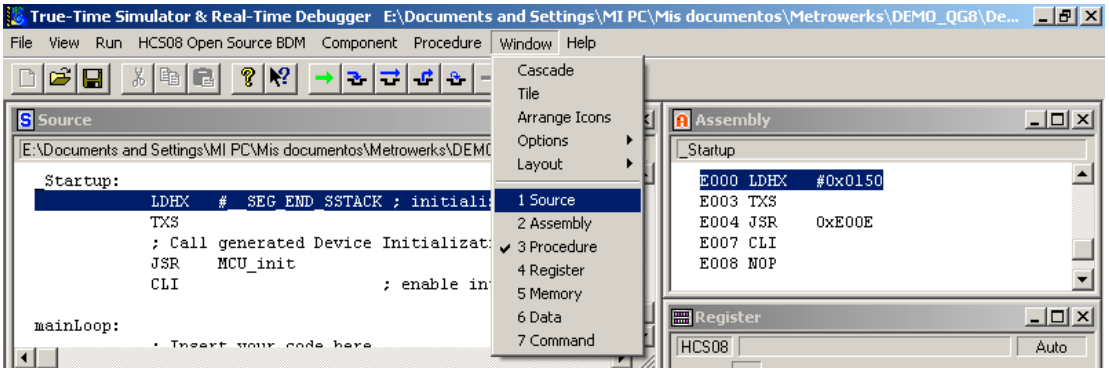

True-Time Simulator & Real-Time Debugger E:\Documents and Settings\MI PC\Mis docu...
File View Run HCS08 Open Source BDM Component Procedure Window Help
Source
Assembly
LOADER WARNING
The debugger is going to mass erase the non volatile memory (eeprom and flash) of the current device, then program the application.
OK
Abort
Do not display this message anymore for this project.
DHX #0x0150
XS
Data
SP FF
Command
in>
Memory
0080 C6 B3 D7 BA ....
For Help, press F1
Automatic (triggers, breakpoints, watchpoints, and trace possible)
MC950

Se abrirán múltiples ventanas, y en la última nos indicará que la memoria FLASH del MCU será TOTALMENTE borrada (mass erase), para luego programar la aplicación a depurar en la memoria del MCU de la placa de "Demo".

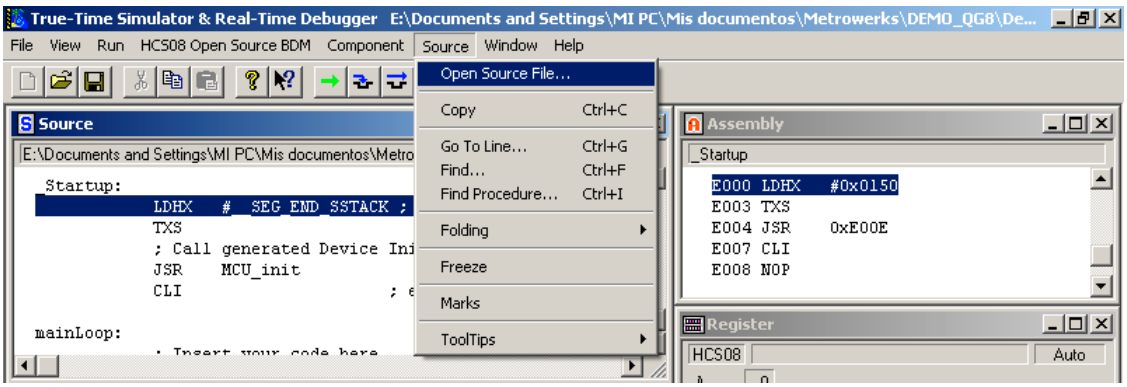
Al hacer click en **"O.K."**, se presentará la siguiente ventana de Debug.....

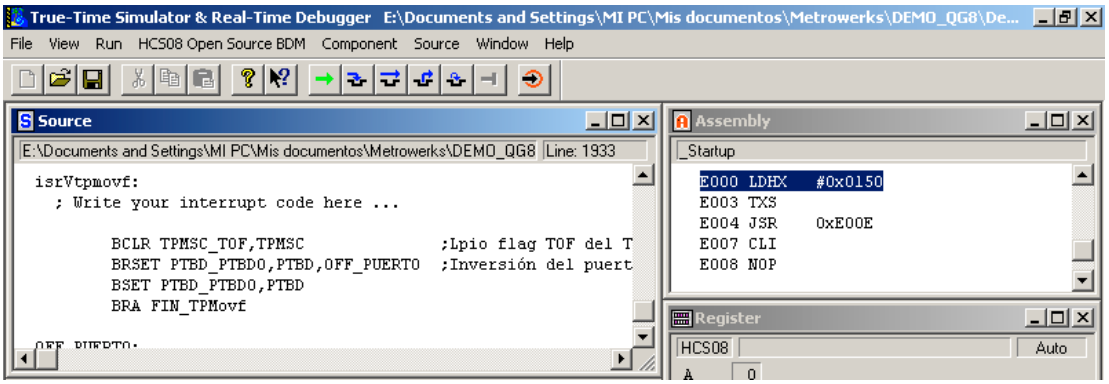


Como se quiere observar el código implementado para el ejemplo, y en especial las líneas agregadas para invertir el estado del puerto PTB0, se deberá seleccionar la opción **"Window"** y al desplegarse la ventana de opciones, se hará click en **"Source"** para activar esta posibilidad.....

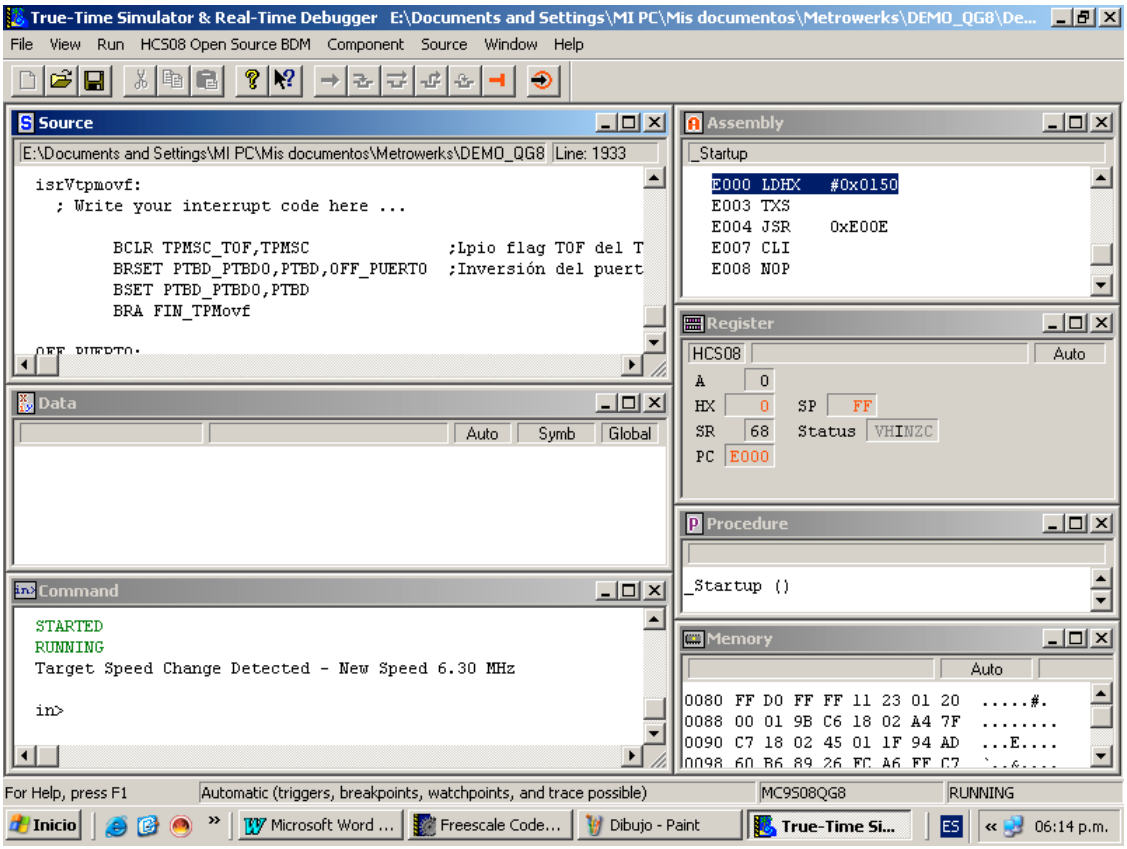


Al activarse la opción "Source", hacer click en **"Open Source File"** para luego elegir el archivo **"MCUint.dbg"**, según se muestra en las figuras.....





Si se quiere hacer "Correr" la aplicación en Tiempo Real, sin colocar un "Break Point" (punto de parada), simplemente hacer click en el icono con la ***flecha "verde"*** y la aplicación comenzará a correr en tiempo real sin detenerse (simultáneamente desaparecerán los iconos relacionados con la ejecución de código en distintos modos), y solo quedarán los de HALT (detención) y Reset (reset general del MCU).



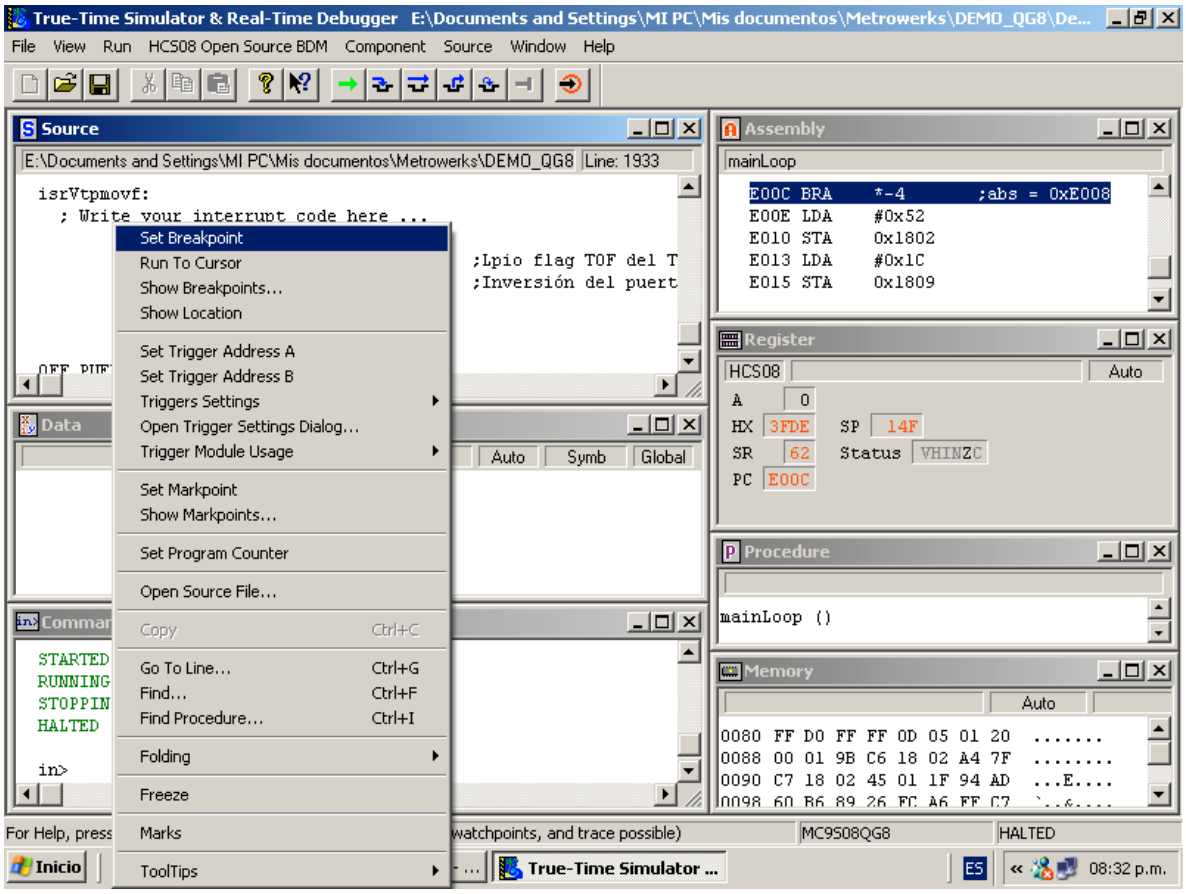
Por el puerto **PTB0** (pin 12 de la placa ***DemoQG8***) saldrá una señal cuadrada de 500 ms de período en forma ininterrumpida .....

Atención:

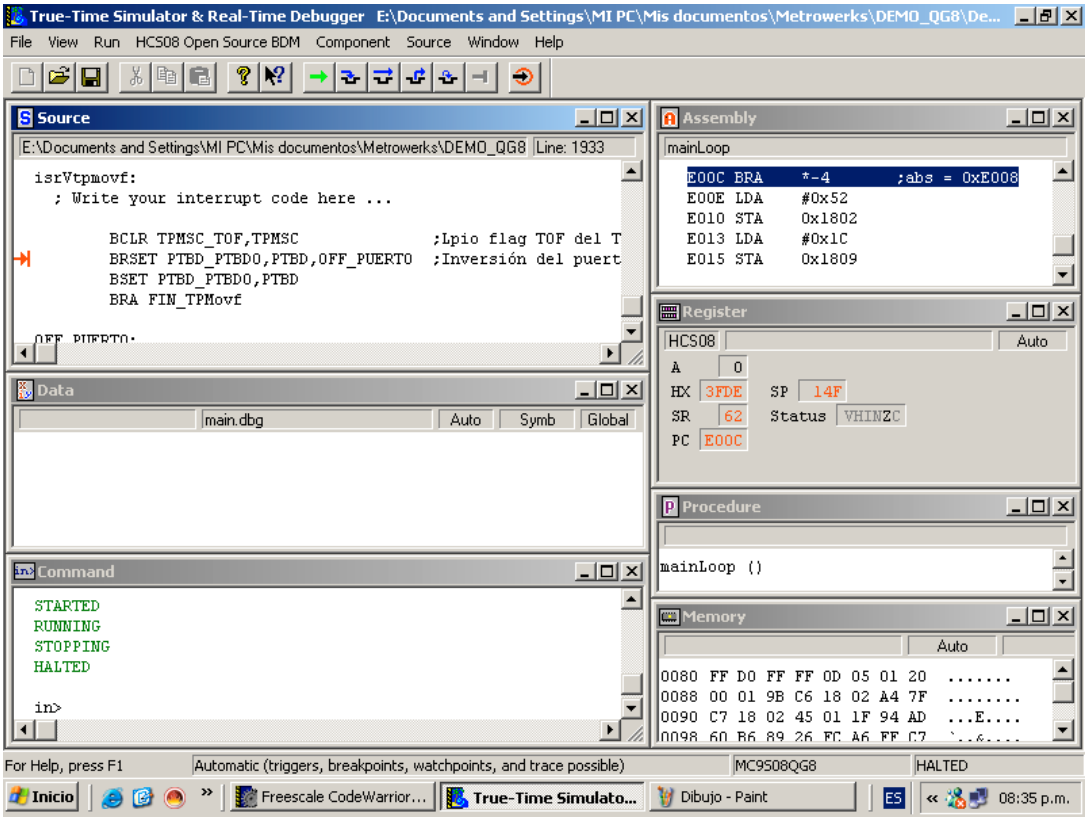
Debido a que el oscilador interno del **MC9S08QG8** se ajusta por medio de unos valores contenidos en la FLASH del mismo (ver sección ICS, Internal Clock Source, en el manual del MCU) y que durante el proceso de borrado de TODA la memoria FLASH los mismos son borrados (con $FF), la frecuencia final del oscilador puede ser DISTINTA a la originalmente configurada en nuestro programa. Hasta el presente **(firmaware R7)** el sistema ***R(S)_POD*** **no puede ajustar en forma automática** el oscilador interno de los HC9S08 / ColdFire V1. Solo el sistema ***USBMULTILINKBDME*** puede hacerlo (ver Addendum 2 : Como utilizar los valores de fábrica de NVFTRIM / NVICSTRIM).

Luego procedemos a detener la ejecución del código de aplicación haciendo click en el icono de **HALT.**....

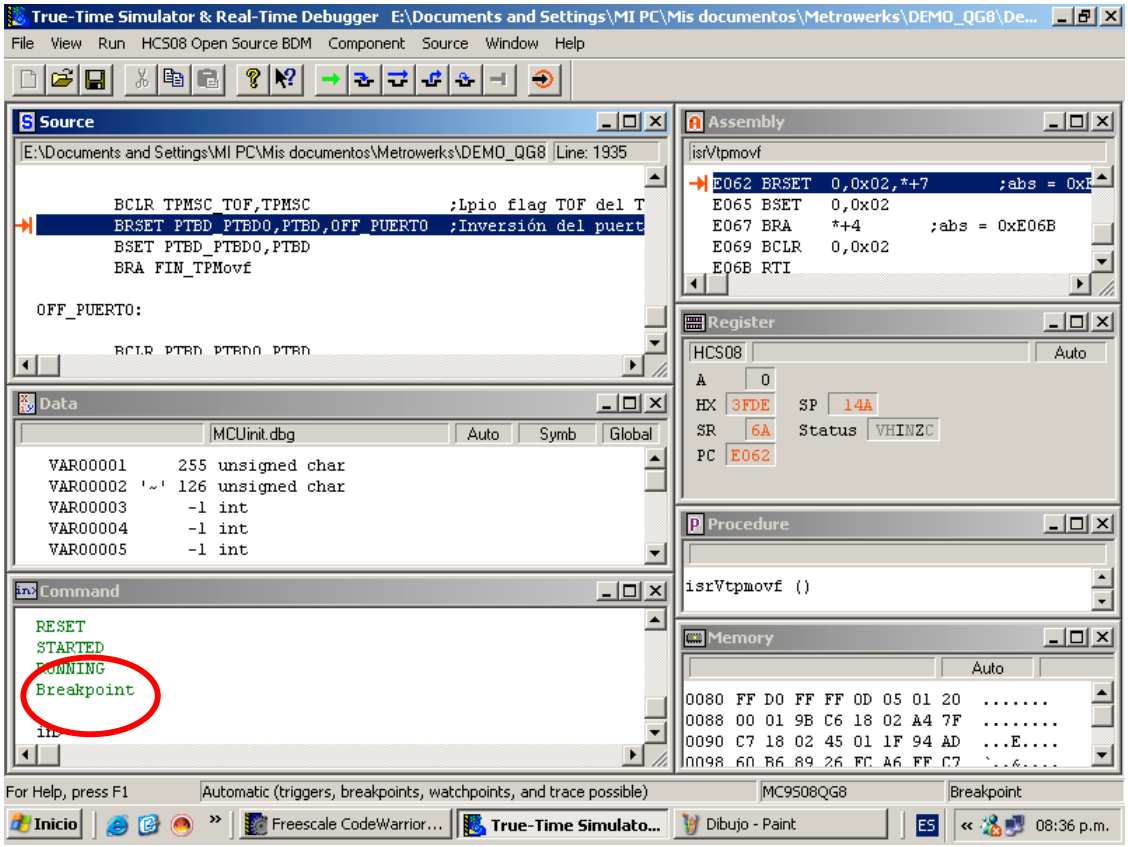
Para colocar un Break Point, en cualquier punto del programa de ejemplo, hacemos click en la línea que nos interesa colocarlo y presionamos el **botón derecho del mouse** y elegimos la opción **"Set Breakpoint"**.



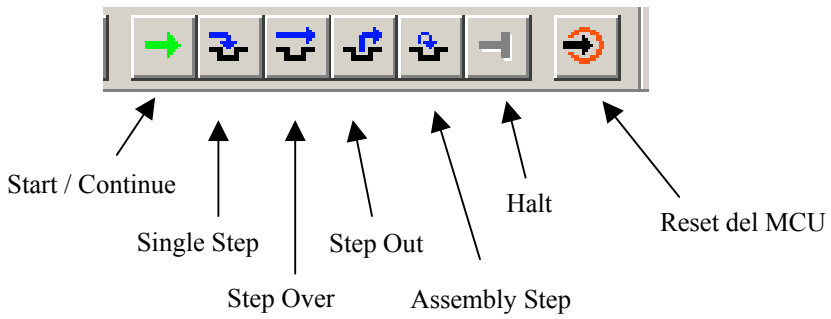
Y se representará el mismo por una **flecha "Roja"** a la izquierda de la línea elegida, según se puede observar en la figura.....



Al hacer Click en la **flecha "verde"** (RUN), la aplicación correrá hasta el breakpoint previamente marcado y se detendrá en el, como se muestra en la figura.....



Sugerimos al usuario, ampliar los comandos disponibles por medio de la opción **"Help"** disponible en el entorno ***"CodeWarrior"*** o en los numerosos tutoriales disponibles en el sitio del mismo.
Solo a título de ejemplo se mencionarán los más importantes disponibles en la barra de herramientas superior del Debugger y ellos son:



- **Start / Continue** → Ejecuta la aplicación en forma continua (RUN) en Tiempo Real.
- **Single Step** → Ejecuta una sola instrucción por vez en tiempo real y luego se detiene.
- **Step Over / Step Out / Assembly Step** → Ejecución paso a paso usadas en distintos Modos de depuración (ver Help).

- **Halt** → Detiene la ejecución del código al presionar el icono y devuelve el control del Debugger.

- **Reset** → Reset del MCU bajo desarrollo (envia comando de reset por pin BKGD y Además fuerza línea RST del conector BDM (Pin 4) a GND por un breve Tiempo (pulso de reset) y luego la libera (útil cuando el MCU dispone del Pin de Reset).

9.0 – Instalación Drivers USB en Windows 98SE.

Tanto los drivers de USB para el control del sistema ***R(S)_POD*** como el entorno ***CodeWarrior 5.0 y 6.x*** han sido diseñados para trabajar en forma natural (nativa) en ***Windows XP***. **Se sugiere al usuario trabajar con este sistema operativo para evitar problemas de funcionamiento, tanto del puerto USB como en el entorno CodeWarrior.**

A continuación se detalla la instalación manual de los drivers USB ya que los mismos no se pueden instalar en forma automática debido a que el instalador no reconoce al **W98SE** como sistema operativo por default.

Copiar los siguientes archivos de la carpeta:

D:\R(S)_POD\INSTALACION R(S)_POD**Drivers**

- **Libusb0.sys** a → C:\windows\system32\drivers\libusb0.sys
- **Libusb0.dll** a → C:\windows\system32\libusb0.dll
- **OpenSourceBDM.dll** a → C:\windows\system32\OpenSourceBDM.dll

Como resultado de esta instalación, al conectar el ***R(S)_POD*** al puerto USB de la PC, el sistema operativo debería reconocer la placa, sin mostrar carteles de "Dispositivo Desconocido" o "Nuevo Hardware encontrado" y en las **"propiedades del sistema"** deberíamos ver que apareciera un nuevo icono bajo el título:

- **Libusb – Win32 Devices → Open Source BDM (LibUSB – W32).**

Es importante reiterar que aunque la instalación se efectúe con éxito, no se puede garantizar que el sistema ***R(S)_POD*** y el ***CodeWarrior*** funcionen correctamente bajo **W98SE.**

10.- Preguntas más Frecuentes.

¿Qué limitaciones posee el sistema R(S)_POD frente al USBMULTILINKBDME?

El sistema ***R(S)_POD*** está basado en el OSBDM (Open Source BDM) desarrollado por la comunidad Freescale y su flexibilidad y compatibilidad estarán en función de las futuras actualizaciones que pudiera tener el mismo, por lo tanto, solo el sistema USBMULTILINKBDME puede asegurar 100% de flexibilidad y compatibilidad con los distintos dispositivos HC9S08 / RS08 / Serie Flexis 8/32 Bits que fueran apareciendo a lo largo del tiempo.

¿Cuáles son las condiciones de equipamiento recomendables para un buen funcionamiento del sistema?

Se recomienda la siguiente configuración para un buen funcionamiento:

- PC o Notebook Pentium IV / Celerón o similar AMD.
- 512 MB o superior en RAM.
- 1 GB libre en disco rígido.
- Conexión ADSL o similar por RED.
- Sistema Operativo Windows XP SP2 o SP3 (Home / Professional).
- Puerto USB 2.0 Host (500 mA corriente).
- CodeWarrior 6.x con las actualizaciones al día.

¿Al conectar la placa R(S)_POD por primera vez al puerto USB la misma no es reconocida como nuevo hardware en el sistema o indica problemas con el dispositivo USB?

Como todo sistema USB, al conectar por primera vez la placa ***R(S)_POD*** al puerto USB, el sistema operativo (***Windows XP***) debe indicarnos la presencia de un nuevo hardware e invitarnos a instalar los Drivers correspondientes. Si ello no sucede, deberemos verificar las conexiones y buen contacto del Cable USB Tipo "A-B" entre la placa y el puerto USB.

- Verificar también que NO se encuentre pulsado (retenido) el switch "SW1" (led LD3 "Power_MCU" encendido) al momento de conectar el cable USB al puerto, ya que ello provocaría un exceso de corriente en el puerto.
- Verificar estado del puerto USB (funcionamiento).
- Verificar estado del Cable USB Tipo "A-B".

¿Qué precauciones hay que tener para trabajar sin riesgos con este sistema?

Como el sistema ***R(S)_POD*** posee la modalidad de proveer alimentación a la placa bajo desarrollo (JP1 en posición 2-3) por medio de un circuito regulador de muy bajas perdidas Conectado al puerto USB de la PC o Notebook en uso, se debe extremar los cuidados de no exceder los límites de corriente máximos de trabajo recomendados, aún cuando el sistema ***R(S)_POD*** posee un circuito especial de protección y limitación de corriente.

- Se recomienda no superar los 300 mA a entregar a la aplicación bajo desarrollo.
- Tener especial cuidado en las tensiones máximas de trabajo del MCU elegido.
- Tener liberado el switch SW1 al conectar o desconectar ***R(S)_POD*** al puerto USB y/o cuando se conecta / desconecta a la placa bajo desarrollo el arnés a CN1.
- Verificar las conexiones en el conector BDM de la placa bajo desarrollo (inversiones, cortos, etc.).
- Verificar que los "jumpers" estén en las posiciones correctas.

¿Cuándo supero los +5V en VDD al ajustar el valor con el preset PR1 se enciende el LED LD4 "OverCurrent"?

Esto es normal, el circuito regulador de tensión indica de esta forma que no puede entregar mayor tensión que la aportada por el puerto USB de la PC o Notebook utilizada.

¿Qué debo hacer con el pin de RESET del conector BDM cuando el MCU no disponga de dicho pin?

El sistema ***R(S)_POD*** tiene incorporado el manejo "Físico" del pin de RESET de los MCUs de la familia HC9S08 / RS08 / Serie Flexis 8/32 Bits, logrando de esta forma el total control del MCU, aún cuando el mismo se encuentre "colgado" sin respuesta en el modo BDM. Existen dispositivos que no poseen dicho pin en forma normal, por lo que se recomienda dejar sin conexión el **"pin 4"** del conector **BDM CN1** en la placa bajo desarrollo en estos casos. De todas formas, el sistema ***R(S)_POD*** enviará un comando "Reset" por medio del pin BKGD al MCU para intentar restablecer las condiciones iniciales al presionar el icono "Reset" en el sistema ***CodeWarrior***, esto puede ser suficiente en la mayoría de los casos, sino liberar / presionar el switch SW1 para lograr un Power On Reset del MCU ante la no respuesta de este.

11.- Actualización de Firmware.

El sistema ***R(S)_POD*** permite ser actualizado (actualización de Firmware) por medio del puerto USB al configurar unos pocos JUMPERS dispuesto en la placa principal.
Esta característica de actualización de firmware (software de la placa) sin utilizar herramientas especiales, permite que ante eventuales mejoras o cambios en el sistema, el usuario pueda efectuar las mismas sin necesidad de recurrir al fabricante.

El sistema ***R(S)_POD*** incorpora un microcontrolador con interface USB MC908JB16, lo que permite la grabación en circuito (ICP) de su memoria FLASH por medio del puerto USB. **Freescale Semiconductor** dispone del aplicativo **"USBICP"** (ver AN2399) que facilita la grabación / borrado de la memoria FLASH de los dispositivos USB y está disponible en el CD ROM de instalación del sistema ***R(S)_POD.***

Para efectuar la actualización de Firmware de la herramienta procederemos de la siguiente forma:

1) Obtener del sitio web de **EduDevices.com.ar**

www.edudevices.com.ar

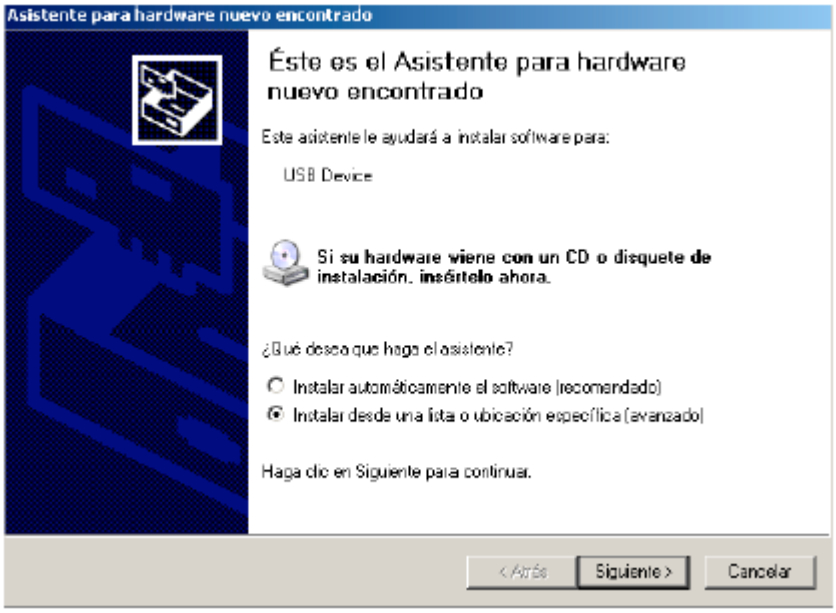
Las actualizaciones de Firmware (**"R5"** hasta Agosto 2008, **"R7"** a partir de Septiembre 2008), si las hubiera, y guardarlas en alguna carpeta de la PC en uso.

2) Con la placa principal de ***R(S)_POD*** sin energía (sin conexión al puerto USB de la PC), sin la placa ***DemoQG8***, y con el Switch **SW1** liberado (VDD → Off). efectuar un "puente de estaño" en el "jumper pad" **JP5**, dejando sin modificar los jumpers **JP4** (Abierto) y **JP3** (en posición 1-2) (**JP1** nunca se utilizará en la actualización).

3) Conectar la placa en esas condiciones al puerto USB de la PC por medio del cable Serial USB Tipo "A – B". Se escuchará un "Ding – Dong" característico del sistema Windows XP indicando la detección de un nuevo dispositivo conectado al mismo.

4) Elegir la opción **"Instalar desde una lista o ubicación específica (avanzado)"** y al hacer click en el icono **"Siguiente"**, elegir la ubicación donde se encuentren los drivers para la grabación ICP por USB que están contenidos en el CD ROM de instalación:

D:\R(S)_POD\INSTALACION R(S)_POD\Actualizacion Firmware

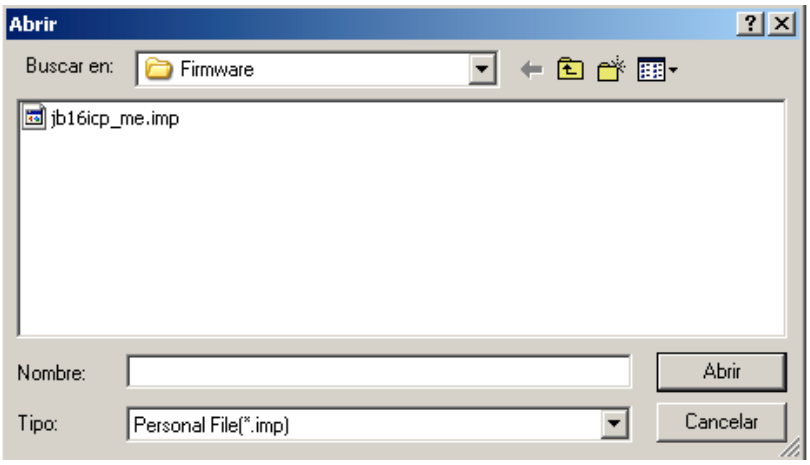

Asistente para hardware nuevo encontrado
Éste es el Asistente para hardware nuevo encontrado
Este asistente le ayudará a instalar software para:
USB Device
Si su hardware viene con un CD o disquete de instalación, insértelo ahora.
¿Qué desea que haga el asistente?
Instalar automáticamente el software (recomendado)
Instalar desde una lista o ubicación específica (avanzado)
Haga clic en Siguiente para continuar.
< Atrás
Siguiente >
Cancelar

5) Luego de unos instantes de búsqueda de los drivers, el asistente de instalación nos indicará que la misma ha concluido con éxito y el nuevo hardware ha sido instalado. Nos dirá que instaló los drivers para ***"Motorola JB16/JG16 ICP Device"*** (necesarios para el borrado / programación en circuito) y se procederá a hacer click en **"finalizar".**

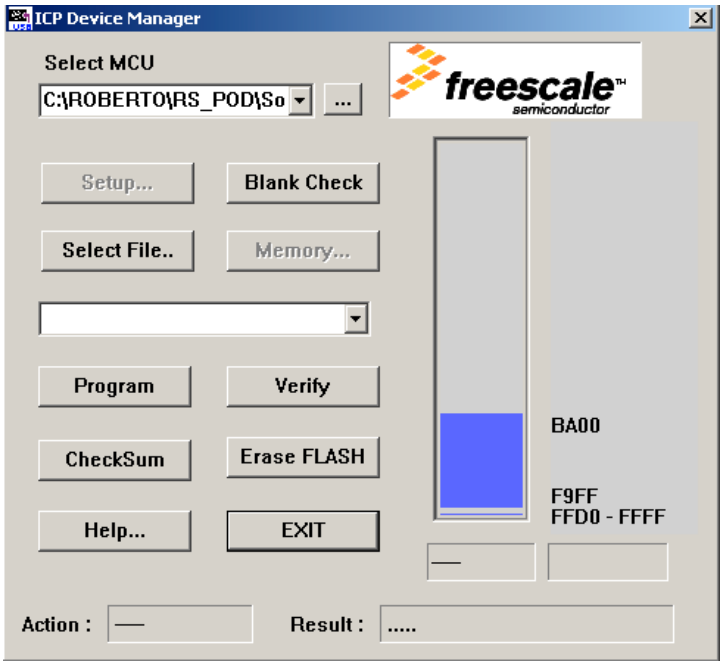
6) Ahora se procederá a "borrar" la memoria FLASH ejecutando el programa **"USBICP.exe"** contenido en la carpeta:

D:\R(S)_POD\INSTALACION R(S)_POD\Actualizacion Firmware

7) Al ejecutar dicho programa, se nos abrirá una ventana que nos pedirá un archivo con extensión "imp", seleccionar el archivo **"jb16icp_me.imp"** como se vé en la figura.



8) Luego de unos instantes, se abrirá una pantalla, como se ve en la figura y hacer "click" en el icono **"Erase FLASH"** para proceder a borrar toda la flash del MCU.



9) El programa nos mostrará que ha finalizado con éxito el proceso de borrado por medio de una pantalla emergente y a continuación salir de la aplicación haciendo "Click" en el icono **"EXIT"**.

10) Proceder a desconectar del cable USB a la placa principal y remover la gota de estaño del jumper **JP5** (Abierto)(verificar que los pads queden perfectamente limpios). Configurar los jumpers JP3 y JP4 de la siguiente manera:

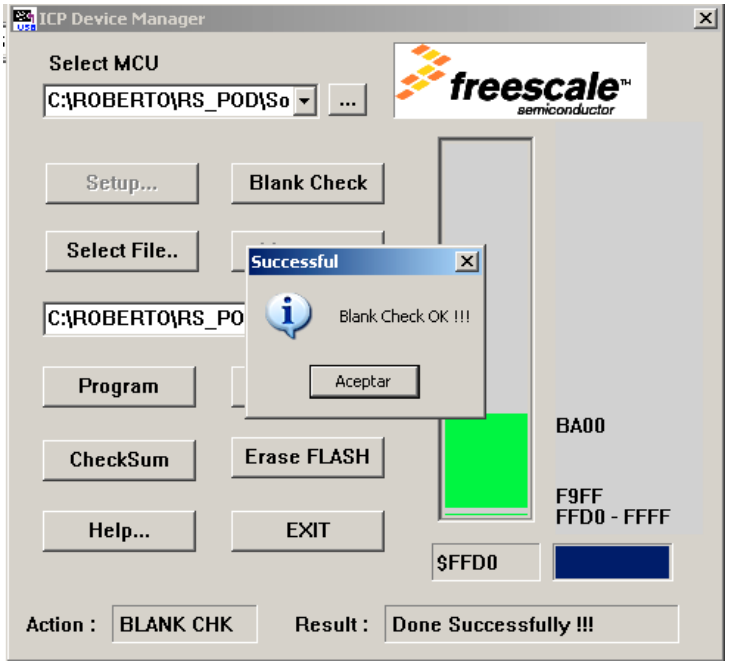
- JP3 → Posición 2-3 (ICP activo).
- JP4 → Cerrado (gota de estaño entre pads) (ICP activo).

11) Volver a conectar la placa al puerto USB y ejecutar nuevamente el programa **"USBICP.exe"**, elegir nuevamente el archivo **"jb16icp_me.imp",** una vez dentro de la pantalla principal, hacer click en la opción **"Select File"** y ubicar el archivo de actualización del Firmware (con extensión ".S19") en la ruta correspondiente.

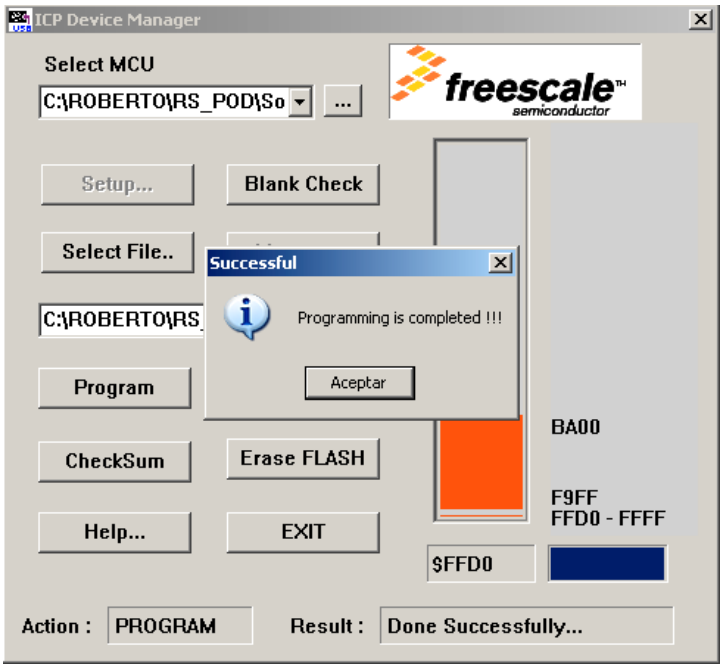
Firmware "R7" →

D:\R(S)_POD\instalacion R(S)_POD\Actualizacion Firmware\Firmware_R7

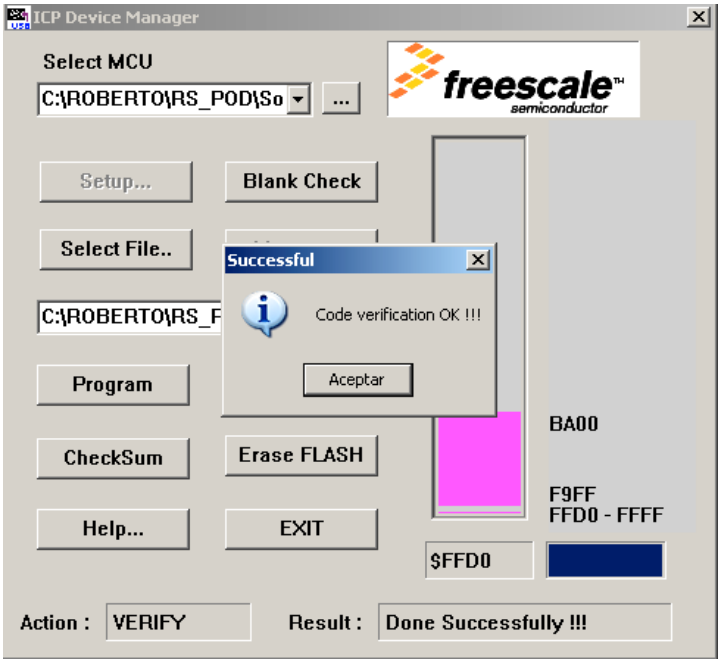
12) Verificar si la memoria ha sido borrada haciendo "Click" en el icono **"Blank Check"**, y si todo está bien, indicará ello por medio de una pantalla emergente como se vé en la figura.....



13) Proceder a grabar el nuevo Firmware (archivo ".S19") haciendo "click" en el icono **"Program"** y si todo está bien, indicará ello por medio de una pantalla emergente como se ve en la figura.....



14) Verificar lo programado haciendo "click" en el icono **"Verify",** y si todo está bien, indicará ello por medio de una pantalla emergente como se ve en la figura.....



15) Desconectar la placa ***R(S)_POD*** del puerto USB y proceder a remover el puente de estaño (gota) del jumper **JP4** y configurar a **JP3** en posición 1-2 para que el sistema quede configurado como viene de fábrica.

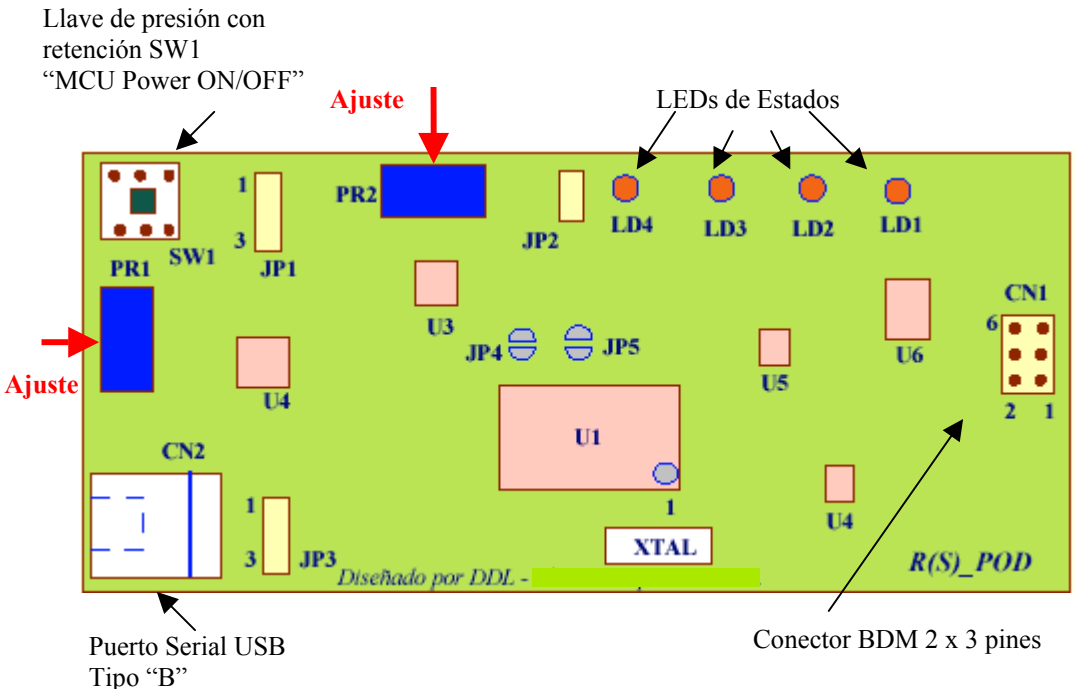
Repetir cada uno de estos pasos para nuevas actualizaciones de Firmware o cuando dentro del proceso de actualización algo no haya salido bien.

NOTA Importante:

Con la versión de **Firmware R7** se deben utilizar los DRIVERS de Windows contenidos en la carpeta **"DRIVERS_R7"** del CD ROM de instalación. Si el usuario dispusiera de un sistema ***R(S)_POD*** con versión de **Firmware R5**, proceder a actualizar el mismo según los pasos indicados anteriormente y luego al conectar el sistema actualizado a la PC***, Windows XP SP2 o SP3*** nos pedirá re – instalar los nuevos drivers para el nuevo firmware, proceder a reemplazarlos con idénticos pasos a los indicados en la **sección 4.2** de este manual.

Addendum 1: Jumpers & Switches Layouts

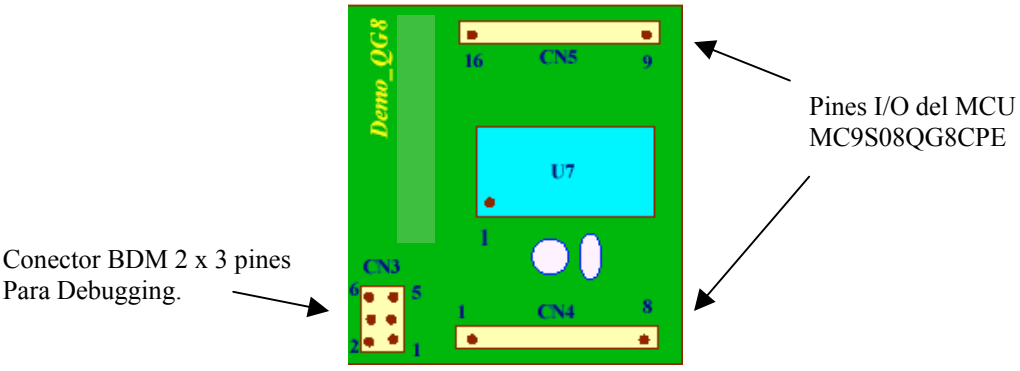
Placa R(S)_POD:

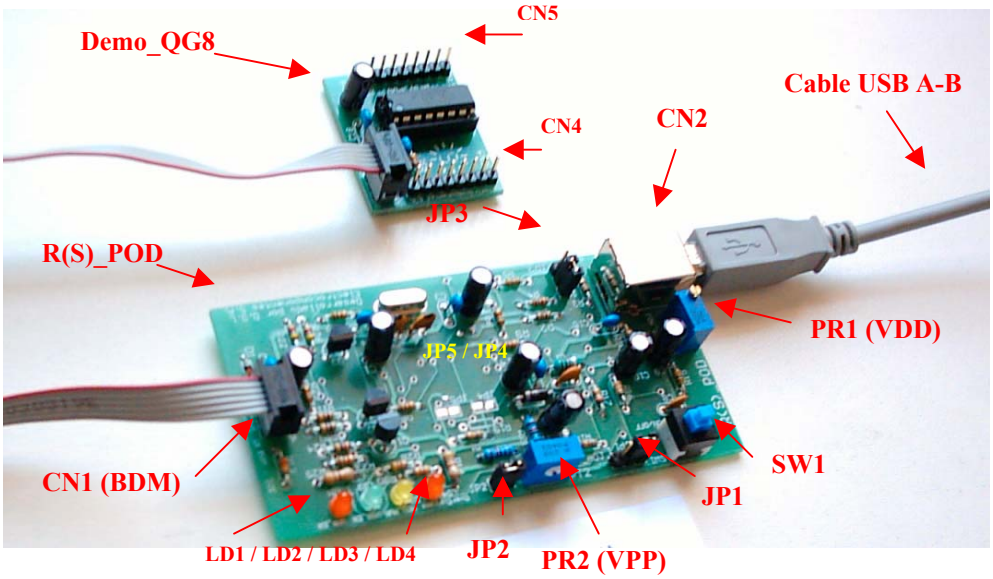


Ajustes:

PR1 → "VDD_ADJ" → VDD ajustable de +1,8V a +5V
PR2 → "VPP_ADJ" → +VPP (ajustar a +12V +/- 0,3V)

Placa *DemoQG8*:




CN5
Demo_QG8
Cable USB A-B
CN4
CN2
JP3
R(S)_POD
PR1 (VDD)
JP5 / JP4
SW1
CN1 (BDM)
JP1
LD1 / LD2 / LD3 / LD4
JP2
PR2 (VPP)

Addendum 2: Como utilizar los valores de fábrica de NVFTRIM / NVICSTRIM

Las familias ***HC9S08, Serie Flexis HC9S08 / V1 ColdFire***, poseen múltiples opciones de osciladores, entre ellas, la opción de utilizar el **Oscilador Interno** (ICS – "Internal Clock Source" o ICG – "Internal Clock Generator") que disponen y que tiene muy buena precisión **(+/- 2%)** si está correctamente ajustado.

Todos los MCUs de las familias anteriormente citadas vienen "ajustados de fábrica" por medio de 2 bytes grabados en posiciones reservadas de la memoria FLASH **(NVFTRIM y NVICSTRIM)** que permiten ajustar la frecuencia de trabajo del oscilador interno simplemente "copiando" estos valores de fábrica a sendos registros en memoria RAM **(ICSSC e ICSTRIM)** dentro del código de programa del usuario (en las primeras líneas de este), de esta forma, el fabricante asegura una buena precisión en los osciladores internos de estos MCUs.

Si estos valores no pudieran copiarse a estos registros, el Oscilador Interno funcionaría con una precisión de **+/- 25%** en todo el rango de temperatura, algo no muy adecuado para aplicaciones que necesiten de buena precisión en los tiempos como pueden ser las comunicaciones Seriales Asincrónicas (SCI) o demoras de tiempo críticas.

Los usuarios del sistema de desarrollo **"R(S)_POD"** se encuentran con el inconveniente de que la herramienta no posee el ajuste automático del oscilador interno **(Auto – Trimming)** como si lo hace el sistema USBMULTILINKBDME.

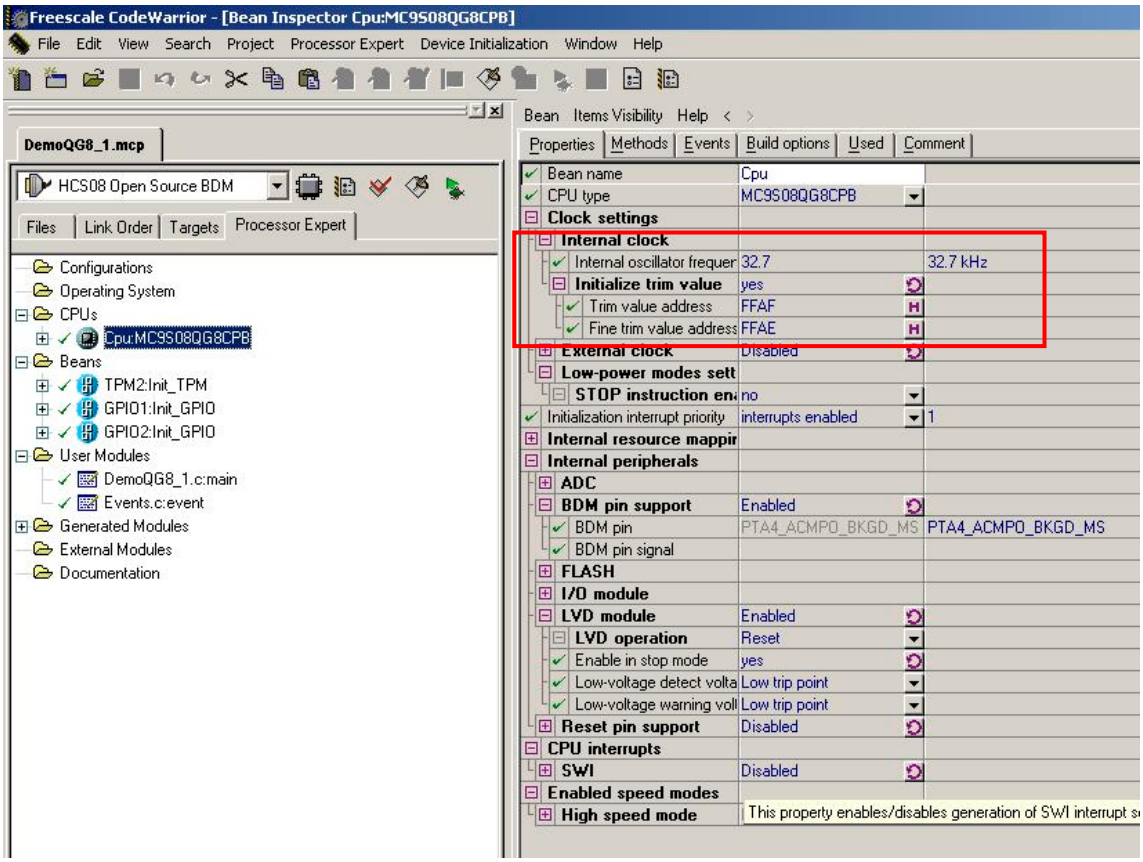
Para compensar este inconveniente, el usuario puede utilizar los valores contenidos en los registros NVFTRIM y NVICSTRIM ubicados en la zona de memoria FLASH y grabados durante el ajuste automático que se hace de fábrica del oscilador interno de los HC9S08.

Este ajuste de fábrica asegura una precisión en la frecuencia de +/- 2% y ello es más que suficiente para la mayoría de las aplicaciones que utilizan este tipo de oscilador.

Para conservar los valores de fábrica grabados en FLASH, el usuario deberá evitar el "borrado en masa" (mass erase) de la memoria FLASH, por lo que esta serie de indicaciones solo serán válidas para aquellos dispositivos que se encuentren "virgenes" de fábrica, una situación más que común durante la etapa de producción de equipos y sistemas.

Pasos a seguir:

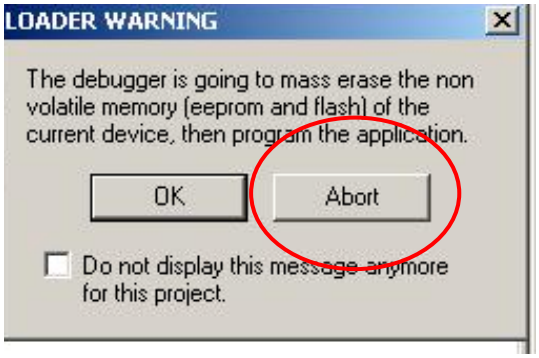
1) Cuando se utiliza la aplicación ***"Processor Expert"***, configurar el bean "CPU" para habilitar la función **"initialize trim value"** que permite utilizar los valores almacenados en posiciones de flash y copiarlos a los registros de RAM. En nuestro ejemplo, el MC9S08QG8 dispone de dichos valores en las posiciones de flash $FFAE y $FFAF.



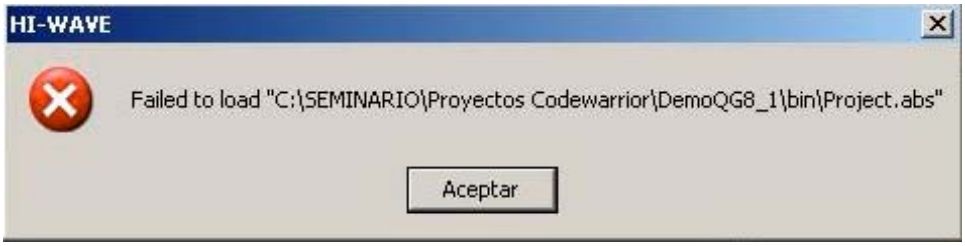
2) Una vez generado el código con el "Processor Expert", en el archivo **"MCUint.asm"** se generarán en forma automática las líneas de código que permiten copiar los valores almacenados en las posiciones de flash grabadas de fábrica, en los registros en RAM **ICSTRIM** e **ICSSC**, como se puede ver en la siguiente figura.....

```
        AND     #$CF
        STA     SPMSC3
;  System clock initialization
        ; ICSTRM: Initialize internal clock trim from a non volatile memory
        LDA     $FFAF
        STA     ICSTRM
        ; ICSSC: Initialize internal clock trim from a non volatile memory
        LDA     $FFAE
        ORA     #$FE
        STA     ICSSC
        ; ICSC1: CLKS=0,RDIV=0,IREFS=1,IRCLKEN=0,IREFSTEN=0
        MOV     #$04,ICSC1              ; Initialization of the ICS control register 1
```

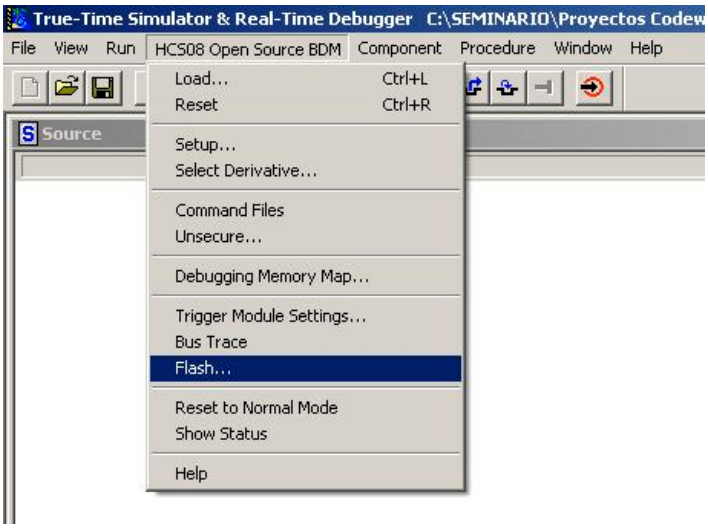
3) Una vez compilado el código, se está en condiciones de grabar la memoria flash de programa del MCU y/o proceder a la Emulación en Tiempo Real por medio del **"Debugger" (Hiwave)** del entorno CodeWarrior. Como se ha visto anteriormente, ello se consigue al hacer "click" en el ícono con la **"flecha verde"** de la barra de herramienta del editor. Al abrirse el entorno de Debugging, el mismo preguntará por la opción de **"Borrado en masa" (mass erase)** de la memoria FLASH, proceder a "ABORTAR" el borrado en masa para conservar los valores de fábrica, según se muestra en la figura.


LOADER WARNING
The debugger is going to mass erase the non volatile memory (eeprom and flash) of the current device, then program the application.
OK
Abort
Do not display this message anymore for this project.

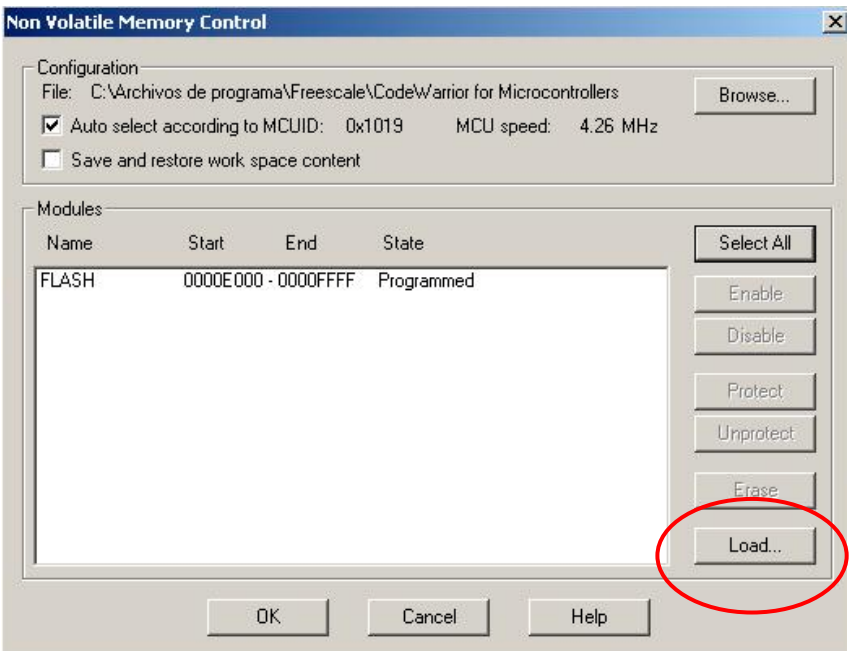
4) Como resultado de esta acción el sistema **nos indicará un estado de error de carga del programa en la memoria FLASH del MCU** (el mismo no se ha cargado aún), hacer click en "aceptar" para continuar.....



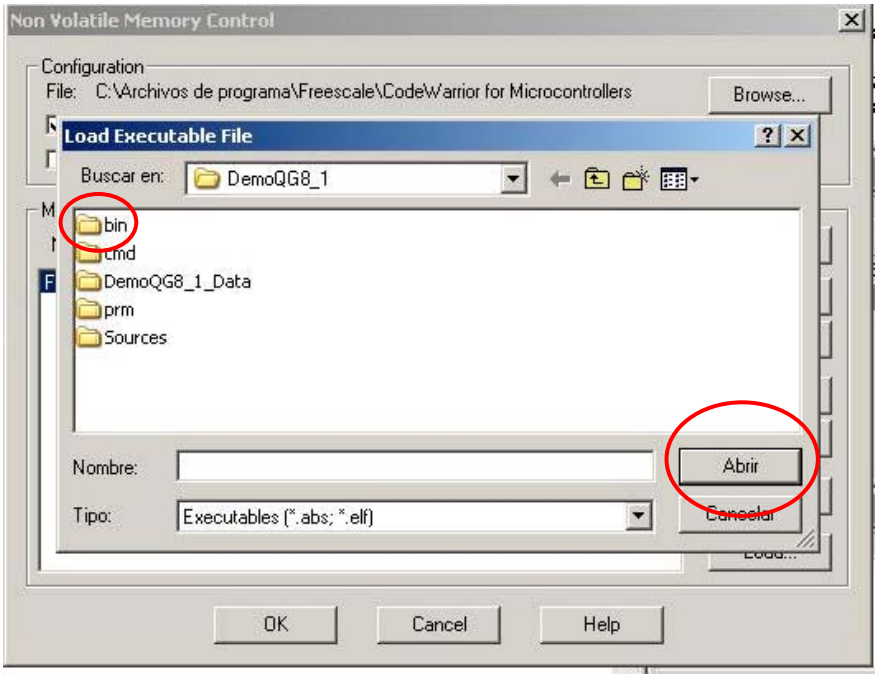
5) Hacer Click en la opción ***"HCS08 Open Source BDM"*** de la barra de herramientas del debugger y elegir la opción **"Flash"** como se ve en la figura.



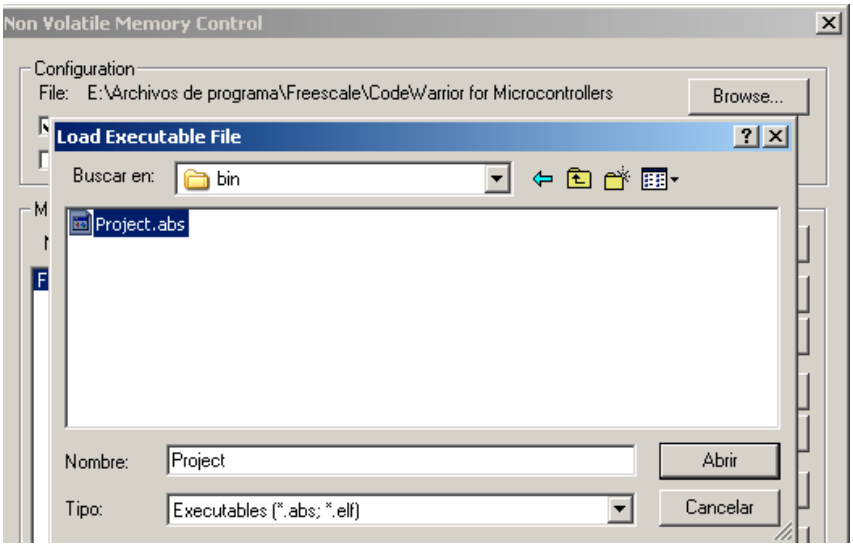
6) Una vez dentro de la opción **"Flash",** hacer "click" en el ícono **"Load"** que nos permitirá cargar el programa (en formato "S19") en la memoria FLASH respetando los valores previamente grabados de fábrica.



7) Al abrirse la ventana ***"Load Executable File"***, elegir la carpeta **"Bin"** que contiene el archivo ejecutable a cargar en la memoria flash del MCU y luego hacer "click" en "abrir" para confirmar la operación.....

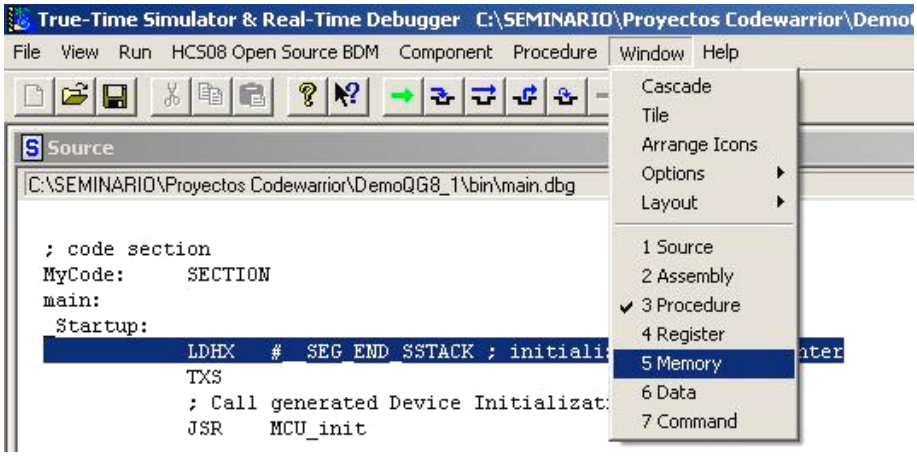


8) Dentro de la carpeta **"bin",** estará el programa ejecutable bajo el nombre **"Project.abs"**, seleccionarlo y luego hacer click en "abrir" para confirmarlo....

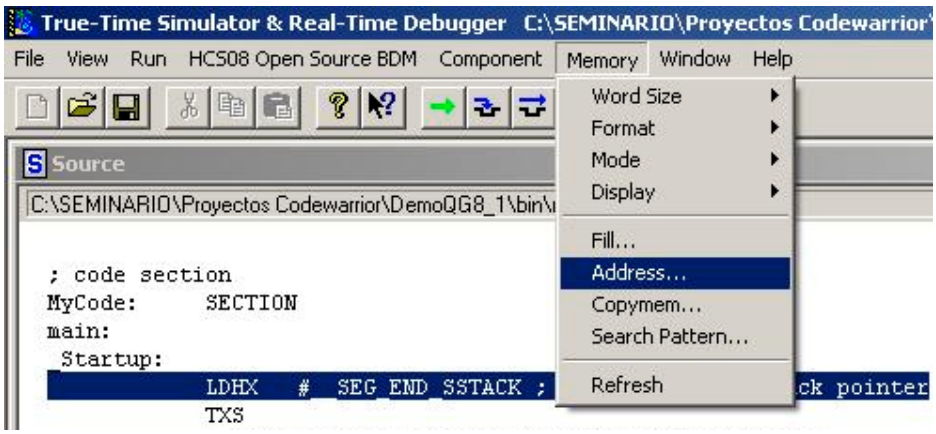


9) Hecho lo anterior, el programa ejecutable del usuario se cargará en la memoria FLASH del MCU sin alterar los valores contenidos en las posiciones reservadas para los valores de ajuste de fábrica del oscilador. Como el programa del usuario utilizará estos valores para copiarlos a los registros en RAM del ajuste del oscilador interno, la aplicación correrá con el oscilador ajustado a los valores de fábrica.

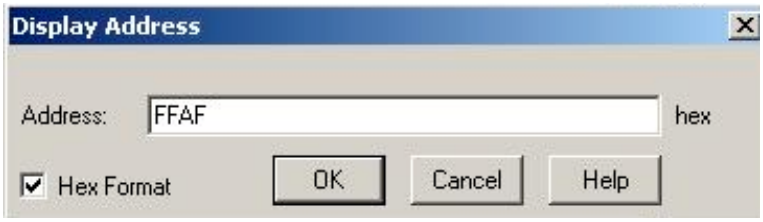
10) Para comprobar que ello es así, el usuario puede observar los valores grabados de fábrica en la memoria Flash haciendo click en la opción "Windows" de la barra de herramientas del entorno de Debugger y luego hacer click en la opción "Memory".



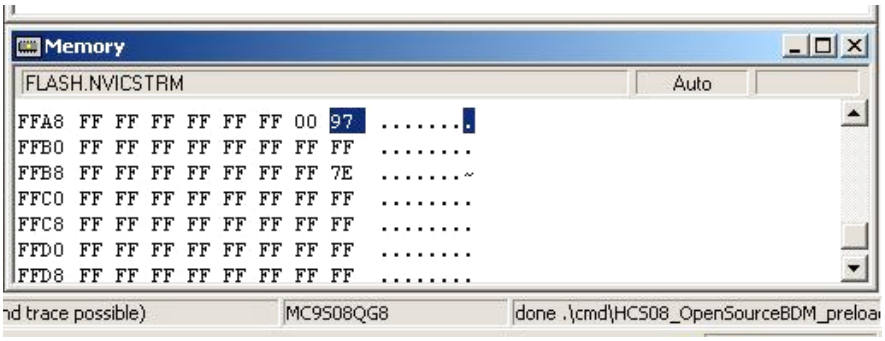
11) Una vez hecho ello, aparecerá en la barra de herramientas la nueva opción "Memory" y alli se deberá seleccionar la opción "Addres" según se puede ver en la figura....



12) Para nuestro ejemplo, donde utilizamos la placa "DemoQG8" que contiene el MCU MC9S08QG8CPE, elegiremos la dirección **$FFAF** que corresponde a la posición de memoria reservada en flash para el valor de fábrica **NVISCTRIM** y **$FFAE** para el valor de fábrica **NVFTRIM.**



13) Finalmente, en la ventana "Memory" pueden observarse los valores de fábrica en las posiciones de flash ($FFAE / $FFAF) para el MC9S08QG8, según se muestra en la figura...


Memory
FLASH.NVICSTRM
Auto
FFA8 FF FF FF FF FF FF 00 97 ....
FFB0 FF FF FF FF FF FF FF FF ....
FFB8 FF FF FF FF FF FF FF 7E ....~
FFC0 FF FF FF FF FF FF FF FF ....
FFC8 FF FF FF FF FF FF FF FF ....
FFD0 FF FF FF FF FF FF FF FF ....
FFD8 FF FF FF FF FF FF FF FF ....
nd trace possible)
MC9S08QG8
done .\cmd\HCS08_OpenSourceBDM_preloa

De esta forma el usuario puede aprovechar los valores de fábrica para correr las aplicaciones con una buena precisión del oscilador interno.

¡Gracias por adquirir el sistema *R(S)_POD*!